DISTRIBUTED TASK ALLOCATION IN MULTI-ROBOT NETWORKS

BY

MOHAMMED MAHMOOD ALI AL-SHABOTI

A Thesis Presented to the

DEANSHIP OF GRADUATE STUDIES

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the

Requirements for the Degree of

MASTER OF SCIENCE

In

COMPUTER NETWORKS

APRIL 2016

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS
DHAHRAN 31261, SAUDI ARABIA

DEANSHIP OF GRADUATE STUDIES

This thesis, written by **MOHAMMED MAHMOOD AL-SHABOTI** under the direction of his thesis adviser and approved by his thesis committee, has been presented to and accepted by the Dean of Graduate Studies, in partial fulfillment of the requirements for the degree of **MASTER OF SCIENCE IN COMPUTER NETWORKS**.

Thesis Committee

Dr. Uthman Baroudi (Adviser)

Dr. Ashraf Mahmoud (Member)

Dr. Anis Koubaa (Member)

Dr.Ahmad Al-Mulhem
Department Chairman

Dr. Salam A. Zummo
Dean of Graduate Studies

29/5/16
Date

عمادة الدراسات العليا
DEANSHIP OF GRADUATE STUDIES

Dedication

To my loving parents and wife.

ACKNOWLEDGMENTS

All praises, all glory and all thanks are due to Allah, The Majestic, The Almighty for bestowing me with knowledge, guidance, patience, courage and health to achieve this work. May peace and blessings be upon prophet Mohammed (PBUH), his family and his companions. I would like to acknowledge the King Fahd University of Petroleum & Minerals and my home university, Taiz University, for the support extended towards my research and providing me the opportunity to pursue graduate studies. I wish to express my sincere gratitude and appreciation to Dr. Uthman Baroudi who served as my thesis advisor, provided me substantial motivation, technical knowledge and philosophical guidance and supported me all the way relentlessly. I also wish to thank the other members of my thesis committee Dr. Anis Koubaa and Dr. Ashraf Mahmoud for their constructive support and encouragement. Very special thanks to my beloved mother and father and to my dear wife, who remains willing to engage with the struggle, and ensuing discomfort, of having a partner who is busy with the studies, research, work and business. My greatest gratitude goes to my friend and roommate Mr. Gamal Sallam, our study and funny times are unforgettable. Finally, I would like to thank all my friends Mohammed Aldarwbi, Yousef, Osama, Taher, Ahmed Al-Areeq, Mohammed Al-Ezy and all my other friends for their support and encouragements and ideas - you guys were more than just friends.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

THESIS ABSTRACT

NAME: Mohammed Mahmood Al-Shaboti

TITLE OF STUDY: Distributed Task Allocation in Multi-robot Networks

MAJOR FIELD: Computer Networks

DATE OF DEGREE: April 2016

Recently, mobile robotics applications are witnessing huge acceptance and penetration in diversified disciplines. The use of the multi-robots system (MRS) has significant improvements over the use of single robot system. Compared to a single robot system, the MRS is time efficient, more flexible and easy to adapt to different applications and scenarios. However, coordinating a team of robots to achieve the desired mission is not a trivial problem; it needs to assign a task to a robot in an optimal way that archives several objectives such as load balancing, quality satisfaction, etc. This problem is called multi-robot task allocation (MRTA) and it is a very challenging problem in the multi-robots system. Many approaches have been developed for MRTA; most of them are centralized and provide offline solutions. The distributed approaches that provide efficient solution suffer from high computational requirements, such as a combinatorial auction. Moreover, the *traveled distance is the only factor considered in the majority of MRTA. In this thesis, we consider the scenarios where there is no a priori information about the tasks that appear dynamically and need to be assigned in a distributed fashion considering the common objectives for MRS applications: total traveled distance, load balance among available robots, task quality satisfaction, and available energy and resources in a robot. Satisfying these objectives will lead to an efficient MRS system. The challenge is to obtain an efficient solution for such competing objectives. We propose tow distributed task allocation approaches. The first one is based on the auction paradigm, and it has two flavors for combining these objectives; the first one uses the weighted sum model (WSM) while the second one utilizes the fuzzy logic. A novel method has been proposed for the first approach to consider tasks synergy in dynamic scenarios. We also proposed and study a new multi-objective threshold-based approach. Extensive simulation experiments, using Webots simulator, have been conducted to study the effectiveness and the applicability of the proposed approaches using different performance metrics. The results show that the proposed approaches have satisfied the objectives at the expense of an increase in the traveled distance compare to the minimum distance auction approach. We have also illustrated our approach using a real experiment using three Turtlebot2 robots in a cleaning like a scenario, in which we show how easily to transfer the proposed solution into a real robotics network. The results from the experiment have a similar trend to the ones we have obtained from the simulation, which implies the practicality of the proposed approaches.*

ملخص الرسالة

الاسم الكامل: **محمد محمود علي الشبوطي**

عنوان الرسالة: **إسناد المهام لا مركزياً في شبكات الروبوتات الجماعية**

التخصص: **شبكات الحاسب**

تاريخ الدرجة العلمية: **نيسان 2016**

في الآونة الأخيرة, تزايدت استخدامات الروبوتات المتحركة بشكل ملحوض في مختلف المجالات. كما أن استخدام نظم الروبوتات الجماعية له عده ميزات مقارنة بنظم الروبوت الواحد حيث أنها ذات كفاءة أفضل فيما يتعلق بالوقت, بالإضافة لسهولة ملائمتها لمختلف التطبيقات والسيناريوهات. لكن عملية التنسيق وتوزيع العمل بين الروبوتات داخل أنظمة الروبوتات الجماعية ليس بالأمر الهين, إذ يجب أن تسند كل مهمة للروبوت المناسب بشكل مثالي إعتماداً على عده عوامل مثل توزيع الحمل بالتساوي وتنفيذ المهام بالجودة المناسبة وإلخ. هذه المشكلة تسمى توزيع المهام على مجموعة من الروبوتات وهي تحدي رئيسي في نظم الروبوتات الجماعية. طورت عدة خوارزميات لحل هذه المشكلة, أغلبها تعتمد على الحل المركزي للسيناريوهات المكتملة حيث تعالج كل البيانات في نقطة واحدة مركزياً. ينما الخوارزميات اللامركزية, مثل المزاد المتعدد, والتي تعطي نتائج فعالة تتطلب معالجات ذات تكلفة حسابية عالية. وبشكل عام فإن كل الخوارزميات بشقيها المركزي واللامركزي عند توزيعها للمهام تعتمد على عامل المسافة المقطوعة.

في هذه الأطروحة, نقترح خوارزميتان جديدتان لتوزيع المهام في نظم الروبوتات الجماعية و نستهدف السيناريوهات المتغيرة والتي لا يمكن حلها بالطرق المركزية. حيث أن المهام لا تكون محددة مسبقاً بل تظهر مع الوقت وتحتاج لأن تسند إلى الروبوتات بشكل تلقائي وآني. كما أننا نعتمد في عملية توزيع المهام على الروبوتات على عده أهداف وهي: تقليل المسافة الكلية التي يقطعها الروبوت الواحد, توزيع المهام بشكل متساوي على الروبوتات, تنفيذ المهام بالجودة المطلوبة, آخذين بالإعتبار الطاقة والمصادر المتوفرة في الروبوت. الخوارزمية الأولى مبنية على خوارزمية المزاد المعروفة, وتحسب المزاد لكل مهة بطريقتين: 1) المنطق الضبابي. 2) نموذج الجمع الترجيحي. والخوارزمية الثانية مبنية على خوارزمية الحد الأقصى. وكلاهما تأخذان بالإعتبار التآزر بين المهام وكذلك الجودة المطلوبة لكل مهمة .

تمت محاكاة الخوارزميات المقترحة بواسطة برنامج المحاكاة Webots وأظهرت النتائج أن الخوارزميات المقترحة أسندت المهام للروبوتات اعتماداً على الأهداف المطلوبة مع زيادة بالمسافة المقطوعة لكل روبوت عنها إذا كان الهدف هو تقليل المسافة المقطوعة فقط. بالإضافة لذلك فقد تم تطبيق خوارزمية المزاد على روبوتات حقيقية (Turtlebot2) وأظهرت التجربة سهولة تمثيل الخوارزمية على روبوتات حقيقية. النتائج التي حصلنا عليها من التجربة العملية طابقت نتائج برنامج المحاكاة, مما يدل على واقعية الخوارزميات المقترحة.

CHAPTER 1

INTRODUCTION

Robotics network gets more and more proliferation in industrial and scientific applications. It attracts researchers attention, because of its computational, sensing, communications and movement capabilities. The market of robotics is growing rapidly over the past five years [3], which promises for a wide use of robotics in our future. Therefore, different robotics topics such as mechanics, control, perception, artificial intelligence and interactions all need to cope up with the future challenges of robotics discipline. The most needed of robotics applications is where human intervention is limited or denied such as search and rescue operations, surveillance, logistic and humanitarian demining, it could be also used for applications where there are economic benefits for using mobile sensors such as farming or production line applications. Advantages of using robotics network are including, but not limited to, the flexibility of modifying the robotics network to match different application scenarios, the robustness of multi-robot system against failure and parallelism operation, which leads to time efficient system.

1.1 Multi-Robot System

Multi-Robot System (MRS) is a distributed system consists of a number of individual computational intelligent system [4]. MRS used in the applications where it is difficult or impossible to use individual robot/agent. Applications of MRS include information acquisition, disaster management and rescue operations [5], remote sensing, coverage and exploration [6] etc. The benefits of using MRS over the individual robot system are, [7]:

- Some tasks inherently require multi-robot such as soccer game.
- Task can be accomplished in a short time using a team of robots.
- The multi-robot solution is more flexible and adaptive to many applications.
- Due to redundancy in the multi-robot system, it is more robust and immune to failure.
- Localization is more accurate if multi-robot are used are instead of a single robot.
- Designing a team of simple robots is easier than designing a single complex robot to do all the work alone.

Therefore, there are various types of application domains where multi-robot system is used such as:

- *Search and rescue operations* in which human intervention is limited or denied due to the danger of the mission itself. Therefore, a team of robots autonomously deployed to do the mission [8].

- *Remote operation* where a team of robots remotely used to do the mission for example border monitoring [9].

- *Intelligent environment* where robotics will play a main role to bring intelligent environments into offices, schools, hospitals, etc [10].

- *Automated construction* especially for a large-scale construction where heavily lifting capabilities required [11].

- *Education and entertainments* involves robots for either educational purposes or for entertainments. Robots need to cooperate in such applications such as a soccer team [12].

- *Agricultural robots* have been used for crops harvesting and for others agricultural activities, and there are already some companies used such team of robotics [13].

Mainly a team of robots used to accomplish certain tasks (system mission) autonomously, they do that better if they cooperate with each other. The tasks are either explored and located by robots themselves or by an external subsystem such as Wireless Sensor Network (WSN), remote sensing system. A task is abstract for the operation performed by the robot, it could be locating the source of a gas leak in a rescue operation, cleaning the floor in a cleaning operation scenario, or crop harvesting in an agricultural scenario.

1.2 Multi-Robot Task Allocation (MRTA)

Multi-robot task allocation (MRTA) is at the core of MRS challenges; it answers the question how to assign tasks to robots in an efficient way considering a set of metrics such as energy, resources, load balance, coverage area [6] or exploring time [14].

There are various types of approaches to deal with MRTA,and they can be classified into three main types based on how does tasks' and robots' information collected and processed: a centralized approach where all information about the current status of the system (MRS) is available in a central point. It is suitable for static scenarios, and it suffers from computational exposure when the number of robots and tasks increase. It inherits the drawbacks of centralized solutions such as a single point of failure, high communication overhead, and it responds slowly to local changes [15]. A second type is the fully distributed approaches like threshold-based algorithms (e.g. [16], [17]). These approaches, on the other hand, are robust to failures, flexible, and require fewer computations and communication resources, however, local optimal solution not necessary aggregate to produce global optimal solution thus they yield suboptimal solutions. The third type of MRTA approaches called market-based, it also called auction-based approach. It has desirable features, such as the robustness, scalability and its adaptive to the objective function. Therefore, it has been used widely within the robotics research community. It is considered to be in a mid-way between fully centralized and fully distributed approaches. In market-based approaches, information processing and computation take place within the robots team themselves. It is centralized in the sense that information collected and processed by one robot. It is distributed in the sense that the auctioneer could be any robot and there is no restriction for that. There is neither global control nor global data storage and both robots and data are geographically distributed. Gerkey and Mataric [18] introduced a taxonomy of MRTA problem based on the three features: whether a robot is capable of performing single or multiple tasks at a time it is single-task (ST) or multi-task (MT). The second feature is whether a task requires a single robot (SR) or multi-robot (MR) to be accomplished, and the third feature considers the dynamic or static environment of the problem; it is time-extended assignment (TA) for dynamic scenarios where tasks appear over time, or instantaneous assignment (IA) if the scenario is static and all information for the tasks are available beforehand.

1.3 The Market-based Task Allocation

In most real world scenarios, tasks randomly appear in a dynamic environment, hence offline (before deployment), task assignment is not a feasible solution. Therefore, in such scenarios a distributed approach is preferable to a centralized approach.

One of the well-known used decentralized and heuristic methods for MRTA is the market-based task allocation. In contrast to emergent cooperation approaches such as threshold-based, in market-based, robots cooperate explicitly through messages exchange. It works in a similar way to auction process in the market, where an auctioneer opens an auction then bidders submit their offers, the auctioneer then will grant the item to the bidder with the highest bid. The auction implemented using a contract net protocol (CNP) [19], which is explained in Fig.1.1. In market-based approach, the bid is computed in each robot as a function of its utility of performing a task. There are mainly two common types of auctions namely, single-task auction, combinatorial auction [15]. For single-task auction, there is one task in the auction and it is granted to the highest bidder. While, in a combinatorial auction, multiple tasks are offered in one auction and bidders can bid to any subset of offered tasks based on robots decision.

The problem of single-task approach is that it does not take the synergy between tasks into account, which leads to a suboptimal solution. Synergy is a term used to describe the relationship between tasks; tasks have positive synergy if the total cost of executing them by one robot is less than the total cost if they are executed by more than one robot [15]. For example, if tasks T_1, T_2 are close to each other, and there are two robots R_1, R_2 having the same utility for performing T_1. In a single-task auction, each robot will take one task, which leads to a suboptimal solution. Since tasks are close to each other, then it is more efficient if the single robot takes both tasks. Although the combinatorial auction provides a better solution, in the previous example, it will assign both tasks to one robot, but it requires high computational resources [20]. Sequential single-item auction (SSI) is an alternative to combinatorial auction. In SSI, robots bid on all unallocated tasks and the robot with the smallest bid (if they bid on cost) wins a task, then they will start all over again the bidding process for the remaining unallocated tasks. After the allocation of all tasks, robots will compute the minimum path to their tasks and move accordingly, which is a special case of Travelling Salesman Problem (TSP) [21]. For the previous example, it will provide the same result as the combinatorial auction.

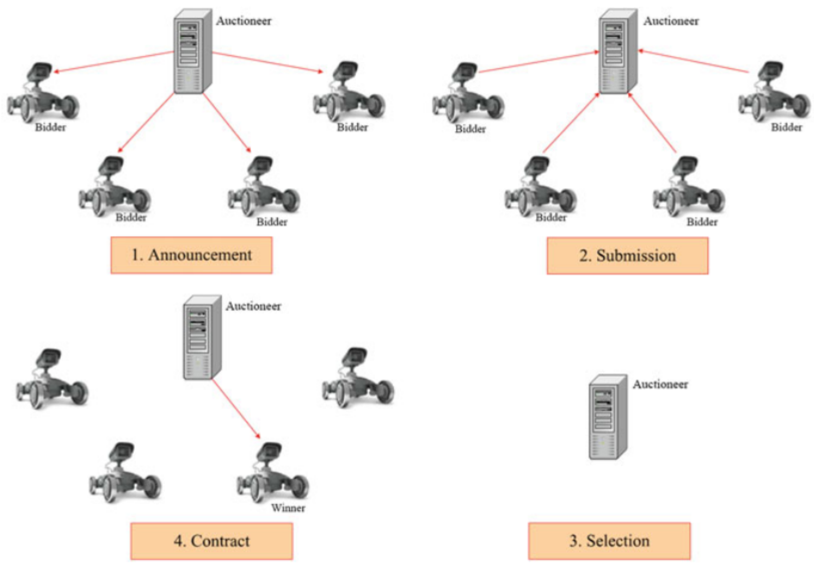


Figure 1.1: Contract net protocol algorithm [1], it has four stages: The auctioneer/coordinator announces the task/s, then each robot computes its bidding value and sends it to the auctioneer in the submission phase, then the auctioneer selects the winner, and finally the winner gets the task.

1.4 Motivation

Motivated by the wide range of applications where the multi-robot system is applicable, and the lack of existing methods that can be applied to dynamic scenarios, where tasks initially are not available, however, they emerge over time. We attempt to solve the MRTA problem in a distributed fashion considering multi-factors in the context of the dynamic scenario. We are targeting task allocation for single task single robot time-extended allocation (ST-SR-TA) systems, where each robot can perform one task at a time and each task requires only one robot. For each assignment process, we consider the following factors: a) available resources and energy to complete the task, such that a task will not be left half executed, b) Traveled distance minimized and the synergy between tasks to minimize the total traveled distance, c) Load balancing among all robots, which will increase the robotics network lifetime, d) Quality satisfaction, such that tasks quality are satisfied and robots quality are utilized. We investigate the relationship between these factors, and our goal is to develop a distributed task allocation algorithm that efficiently assigns tasks to robots, in a dynamic scenario, taking into account minimizing the overall cost based on previously mentioned factors.

1.5 Scope of this Thesis

The scope of this work is MRTA problem in an ST-SR-TA scenarios. Therefore, exploring and locating tasks are out of the scope of this piece of work. Also, we need not to propagate to the stage of task execution and how it should be done, we assume that once a robot reaches a task location, it starts performing the task for a period of time. Further details about the characteristics of the scenarios we are targeting will be discussed later in section 3.3. Also, our goal is not to get the optimal multi-objective task allocation, but rather to investigate auction-based task allocation with multiple factors, and provide simulation and empirical results that show how auction-based works based on these factors.

1.6 Thesis Contribution

The primary contributions of this work are:

- We proposed an online distributed multi-objective auction-based task allocation approach for multi-robot systems based on the auction process. Each objective associated with a metric in which task assignment process depends on. These metrics are:
  - **Workload balance.** A robot with less number of task assignment is preferable to others.
  - **Quality satisfaction.** A robot with a quality level equal or close to the quality level of a task is more desirable than others.
  - **Traveled distance.** The closer the robot the more suitable to execute a task.
  - **Available energy.** A robot with high energy is preferable to others.
  - **Available resources.** A robot with lot of resources is preferable to others.
- We proposed a dynamic method for considering a task's synergy.
- Two methods have been used for combining these factors, weighted sum model, and fuzzy logic system.

- We prove empirically that the two proposed methods (auction-based, threshold-based) can be extended to solve multi-objective MRTA problem.
- We have introduced a quality term, which represents the preferable robot to perform the task.

1.6.1 Quality

There are many applications where it is not enough to distinguish tasks by their type, because there is a vague border between a type of task and another, in such applications we can use, what we call it, task quality. For example, considering high and low-quality video recording as a different type of task will prevent a robot which is equipped with a low-quality camera to execute a task required high-quality video recording. While in some applications you may allow for this if this minimized the traveled distance or the resources of the appropriate robot is busy or to utilize all robots etc. The task quality is a term used as a requirement for the task and the property for a robot. Each robot and task have a level of quality, which represents the match between them. Meaning, a task with low-quality level requires a robot with low-quality level. Although, it seems like a type of tasks however it is more fixable in the sense that a task with high quality is considered to be done even if it has performed with a low-quality robot. For instance, if the tasks are to get an image for a specific location in the terrain, for a classification system, the quality here is the quality of the image. Classification system requires a high-quality image for some objects and medium or low-quality image for others, but anyway, it still can do the classification. Another example, in the inventory the task is to carry an object from place to another, robots have to carry different objects with different weights and size. Quality in such example can be used to prefer a specific robot to carry a specific object, although any other robot can carry it.

1.7 Thesis Structure

The rest of this thesis organized as follows. In the next chapter, we present the related work done to the distributed task allocation in mobile robots network. Chapter 3 we give a description of the multi-objective problem we are considering in this thesis; including the application sample, system model, problem formulation, and the proposed auction-based and threshold-based task allocation. Chapter 4 explains simulation setup parameters and initial settings. Chapter 5 provides the performance evaluation and analysis of the two proposed approaches in the simulation and real experiments . Then we conclude with our major findings and furure directions in chapter 6.

CHAPTER 2

RELATED WORK

The MRTA problem has been studied for different system configurations and scenarios, and there are different centralized and decentralized methods have been proposed. In this chapter, we will give concise related works that have been done in distributed task allocation in multi-robot networks. We will focus on the dominated distributed MRTA methods [22], namely market-based and threshold-based algorithms. In the market-based approach, robots negotiate for a task using auction mechanism, and the auctioneer assigns the task to the highest bidders. In contrast to the auction-based, threshold-based allows each robot to determine by itself without explicit coordination. A robot accepts a task if its "tolerance" surpasses some threshold, otherwise, a task will be ignored. In [23] Karla et. al. have compared between market-based and threshold-based approach under real word condition. Their results indicate that market-based approach is more efficient (with the cost of communication) when information is accurate. In contrast, when the information is not accurate, threshold-based provides same quality task allocation as market-based at a fraction of the cost.

Brian P. et al. [24] proposed MURDOCH, a fault-tolerant distributed protocol based on auction method for multi-robot coordination. The main idea is that the winner gets the task with a contract to finish it within a time window. The Auctioneer is responsible for monitoring the progress of the task, then if it discovers a failure or insufficient progress it can terminate the contract and announces this task in a new auction process.

In [25] Lee, et al. proposed a distributed resource-oriented auction algorithm where they take into consideration the resources that robot consumes while it executes the task. A robot computes an expected cost for each task considering multiple paths for the task. Authors argue that not including this factor can affect the task execution because winner robot may run out of resources while it is executing the task. Also, a multihop action algorithm is proposed for limited robot communication range.

Mi, Zhenqiang et al. [22] they distinguish between discovering and allocating processes. Authors suggest integrating mobile sensors and robots in performing the task discovery and assigning processes. Mobile sensors are responsible for locating and identifying the task while robots are responsible for performing the tasks. When a sensor identifies a task, it sends a request to robots; the closest robot will be the coordinator for that task and will forward the request with a time limit to other robots. Each robot who is in the coordinator's vacancy will participate and computes its utility based on cost, energy, distance, the type of a task and sends the reply. Then, the coordinator will assign the task to the robot with the highest utility. A multihop request will be sent if no robots reply with the participation of mobile sensors network.

W. Sheng et al. [26] used a bidding model for selecting an appropriate robot to discover the unknown area. They include nearness measure, as communication link measure, in the utility function which computes the distance between the current robot and its neighbors. They assume that the high nearness value the high communication links exist between the current robot and its neighbors. The use of nearness measure keeps robots close to each other and reduces total traveled distance and total discovery time.

Elango. et al. [27] proposed a balancing workload by decomposed tasks into clusters based on the total travel distance in each cluster and the distance between tasks on it. Then auction based used to assign robots to the task clusters. However, for clustering tasks, each robot needs to compute its cost for executing all combination of tasks which is a complex process to be done especially if there is a large number of tasks.

In [28], Gong J. et al. proposed a combinatorial auction model based on genetic algorithm (GACA). The GA used for searching on all combination of tasks and robots to get the best solution based on distance. A Hunting task is given as a mission for the team of robots. They have compared combinatorial auction versus single item auction, and they found that in such scenarios combinatorial auction outperforms single item auction with a computational cost.

Gong, et.al. [28] proposed a combinatorial auction based on genetic algorithm, and they test their method in a hunting task scenario. The found that combinatorial auction outperforms the single item auction in terms of time.

Some work has been done in multi-objective task allocation such as the work done by Avraam Th.et al. [29]. They have used auction-based approach to solve multi-objectives which are: remaining energy after task executed, the total time for completion the task, priority of the task. A team of robots is divided into some clusters each of which consists of a coordinator and robots. A coordinator is responsible for arranging the auction and distribute a task information. Robots are heterogeneous and for each task some are eligible to participate in the auction based on the task requirements. A summary of comparing the related works and the proposed approaches are shown in table 2.1.

Reference	Distance	Load balance	Quality	Dynamic scenario	Task synergy
Brian P. et al. [18]	Yes	No	No	Yes	No
Lee, et al. [25]	Yes	No	No	Yes	No
Mi, Zhenqiang et al. [22]	Yes	No	No	Yes	No
W. Sheng et al.[26]	Yes	No	No	No	No
Elango. et al. [27]	Yes	Yes	No	No	Yes
Gong J. et al. [28]	Yes	No	No	No	No
Avraam Th.et al. [29]	Yes	No	No	Yes	No
Propose approach	Yes	Yes	Yes	Yes	Yes

Table 2.1: Summary comparing the related works and the proposed method

2.1 Conclusion

The main contribution of this work is that we assign tasks in a dynamic scenario based on distance, load, and quality satisfaction factors. As well as considering the availability of energy and resources. In contrast to prior works, our approach attempts to study the effects of including all these factors in addition to the newly proposed quality level satisfaction factor. We proposed two methods (the weighted sum model and the fuzzy logic system) for dealing these contradicted multi-factors and come up with a judgment about the appropriate robot to for a task.

In this work we are targeting single-task single-robot time-extended assignment (ST-SR-TA); where each robot is able to perform at maximum one task at a time, and each task requires at maximum one robot, and tasks appear in the environment in a dynamic way.

CHAPTER 3

PROPOSED MULTI-OBJECTIVE TASK ALLOCATION (MOTA) APPROACHES

SR-ST-IA is an instance of the optimal assignment problem (OAP), optimization problem such as Hungarian method gets the optimal solution for such problem. However, in this work we are addressing SR-ST-TA which is an NP-hard problem [18], hence, we proposed heuristic approaches to solve it. The market-based MRTA is the baseline of the first proposed multi-objective task allocation (MOTA). In contrast to the conventional market-based where distance is the only factor for assignment process, MOTA attempts to consider multiple factors (traveled distance, quality satisfaction, load balance, available resources, and energy). These factors are included in the fitness function using weighted sum method to provide a single scalar fitness value. The second proposed approach is threshold-based where robots implicitly cooperate, and they only communicate to solve tie problem (i.e. when two robots are going to execute the same task). We have modified the threshold approach to include multiple objectives as so that it covers a wide range of applications.

Before diving into the details of our proposed approaches, it is a good start to give some applications where the proposed method can be deployed.

3.1 Application Example

Automated farm diseases detection, [30], is an example of applications which our MOTA approaches target. This application includes two systems; remote-sensing and near-range sensing. The aim of the remote-sensing is detecting and diagnosing any unhealthy symptoms in an area of interest such as diseases, weeds, and pests. If any disease detected, remote-sensing advertises a new task for the near-rang sensing system, which is basically a team of robots/mobile-sensors equipped with appropriate sensors such camera, thermography, chlorophyll fluorescence and hyperspectral sensors. The task for a robot is to reach the position where disease symptoms have been detected and gets some images and near-range sensing data such as chlorophyll fluorescence, temperature, humidity, etc. It is also required from the robot to spray fertilizer, potion, pesticides over the infected area. Therefore, the remote sensing system provides a necessary information about a task such as its location, amount of resources required (fertilizers/pesticides), and the quality of the task. In this scenario, quality represents the resolution of the sensors e.g. camera's resolution. Such integration between remote and near sensing systems has been demonstrated a high potential for detecting diseases [31]. The robots may use our proposed MOTA to assign each farming task to a robot in a balanced way considering the traveled distance to the task location, the amount of required resources (e.g. fertilizers, pesticides), as well as the satisfaction of the preferable image quality required (quality level).

Cleanup of factory sites scenario is another example where our approach is applicable. The dust emerges in random places in the area. There is no significant importance to the cleaning order, some are hard to be cleaned (high-quality task) while others are not (medium or low-quality tasks). The mission of the team of mobile robots is to remove dust, litters, statins, etc. Therefore, each cleaning process is considered as a task that should be assigned to an appropriate robot. Tasks information is required to be provided with each advertisement for a new cleaning task including the location of the dust/stain, estimated amount of cleaning chemicals and water resources require, and the preferable quality of cleaning. This information should be provided from an external system, which may consist of a workstation with an array of cameras that monitor the area of interest. Having such system separated from the team of cleaning robots will save the cost of including task detection in each robot [22]. The goal is to map each cleaning task to a robot in a balanced way considering the traveled distance to the dust/stain (task), the amount of required resources (cleaning chemicals and water), as well as the preferable cleaning quality match between a task and a robot.

3.2 Tasks and robots quality

In this section, we introduce in more details the rational behind the concept of quality for tasks and robots. We will consider the scenario of imaging objects for a classification system as a base application for illustrating the concept of quality. Hence, a robot quality is the quality of its camera resolution, while a task quality requirement is the resolution of the image required.

Generally speaking, a quality is a term to be used to represent a preference to pair a robot and a task that requires the same quality level. In other words, we prefer to execute a task with quality level K to be executed by a robot with quality level K and this is considered as the best option. Also, it is possible for a task to be executed by a robot with quality level either higher or lower quality than a task quality requirement. In the first case where a robot quality level is higher than a task quality level, and in this case some extra resources will be wasted from the robot point of view (image storage size, energy consumed by a high quality camera, transmission bandwidth required for a high quality image), while from the task perspective its requirement has been satisfied (the image is captured with a resolution greater than what is required).The other case, when a low-quality robot executes a higher quality tasks, there is no wast in the resource, however, a task requirement does not satisfy 100% (the image has a resolution lower than what is required). To differentiate between a task type and a task quality level, we can say that for the same task type there are different of quality levels.

3.3 System Model

To motivate our MOTA problem, we consider the following system characteristics:

1. The robots R are randomly deployed in the area A of interest.

2. All robots initially have the same amount of energy and resources.

3. The tasks appear uniformly within the area of interest A following Poisson distribution with a mean of arrival λ_t.

4. The order of the tasks' execution is not important; thus a robot is free to perform assigned tasks based on the traveled distance, not on their assigning order.

5. A robot R^i has an exponential distribution service time with a rate $\mu(\, Task/hour)$.

6. A task T^i demands a quality level q_T^i to be performed with.The task quality level is uniformly distributed over the whole range from the minimum to the maximum quality level. The range of quality level is defined by the application itself (e.g. three quality levels would be low, medium and high).

Section 1.6.1 gives a clear description of what quality means throughout this work.

7. A task T^i also requires an amount of resources τ_T^i to be executed, which is uniformly distributed from the minimum task quality requirements to the maximum quality requirements.

8. Task discovering may perform either by a separate system such as mobile sensors or by robots themselves; in both cases, we assume only one robot gets task's details.

3.4 Problem Formulation

A Robotic network composed of m robots can be represents as a graph $G = (R, E)$, where $R = \{R^i : i = 1, 2, 3, ..., m\}$ denotes the set of robots deployed in a 2D area A, $(R^i, R^j) \in E$ $\forall i, j \leq m$ denotes the existing of a communication link between robots R^i, and R^j. A set of n tasks $T = \{T^i : i = 1, 2, 3, ..., n\}$ emerge randomly within the area A follow a Poisson distribution with an arrival rate λ_t $(Task/hour)$. Each task requires at most one robot, and each robot can execute one task at a time. A robot R^i has an exponential distribution service time with a rate of μ $(Task/hour)$, since arrival and service rate are stochastic processes then over a short duration task arrivals for a robot may gets higher than its service rate (i.e. $\lambda > \mu$). Therefore, a robot maintains a list L^i for all arrival tasks. We assume robots are capable of localizing themselves, such as a robot R^i status at a given time represents by its position p_R^i, residual energy E^i, residual resources τ_R^i, number of assigned tasks is the length of L^i ($|L^i|$), and its quality level q_R^i. In the other hand, a task T^i is advertised with its position p_T^i, required resources τ_T^i, and required quality level q_T^i. Task Quality q_T^i can be represented by any number of levels $q_T = \{Q_1, Q_2, Q_3, ..., Q_k\}$, robots also have different number of quality levels $q_R = \{Q_1, Q_2, Q_3, ..., Q_z\}$. Note that robots quality levels have to be distirbuted within the tasks quality level, such that a task always can get a robot with approperiate quality level. Therefore, robots quality levels are driven from the quality levels of the tasks as following:

$$Q_i = (\sum_{j=1}^{j-1} j * \beta) + \alpha \tag{3.1}$$

whereas, β is the range of tasks quality covered by each robots quality, and α is the midpoint of β and they are computed using Equations 3.3, 3.2. Such that a robot covers tasks quality with $\pm\alpha$ above an below its quality level.

$$\beta = \frac{k}{z} \tag{3.2}$$

$$\alpha = \frac{(\beta + Q_1)}{2} \tag{3.3}$$

Whe As a proof of concept, we are going to assume 9 levels of quality for tasks $k = 9$, represented by numbers from Q_1 very low to Q_9 very high-level quality. Using the Equation 3.1 robots' quality has been computed to be $q_R = \{Q_L, Q_M, Q_H\}$:

low Q_L, medium Q_M and high Q_H, whereas $\beta = \frac{9}{3} = 3$, and $\alpha = \frac{3+1}{2} = 2$ and Fig.3.1 illustrates both robots's and tasks' quality.

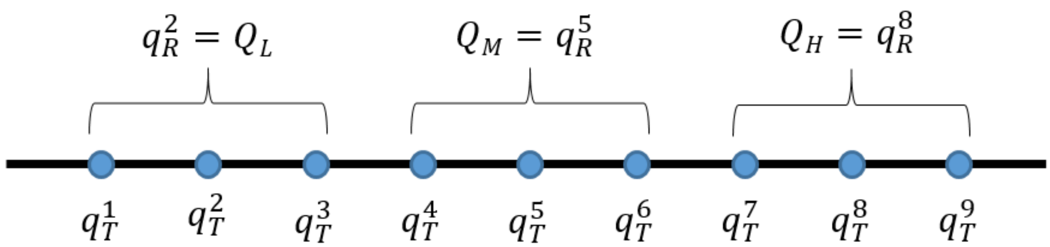


Figure 3.1: Optimal quality satisfaction achieved when a robot performs tasks with quality level close to its own.

Fig.3.2 displays the detailed data for a robot and tasks in the area of interests during task allocation process. Robot R^1, with quality low ($q_R^1 = 2$), has full energy E^1 and resources τ_R^1, and its task list L^1 is empty (i.e. no task has been assigned to this robot). In contrast, robot R^2 has already executed one assigned task T^1 as it is shown in its task list L^2, hence robot R^2 resources and energy level has declined.

3.5 Task Discovery

The proposed MOTA approaches do not include task discovery process. However, we assume that a task is discovered and located at the time it appears using a separate system. Then it is reported to the robotics network along with its requirements including quality level and resources.

This phase can be done in many ways including the following:

- A static wireless sensor network (WSN) can be deployed in the area of

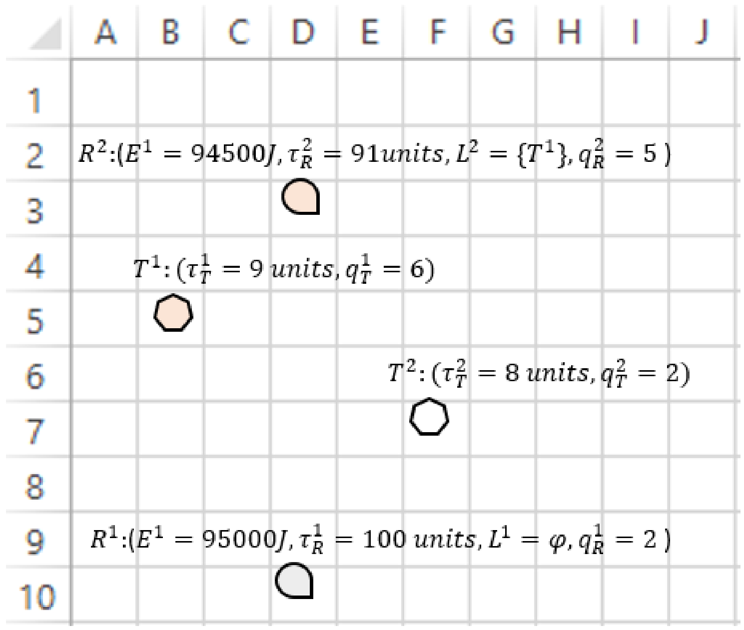


Figure 3.2: Details associated with each robot and task.

interest to discover the tasks.

- A mobile sensor network can be utilized to continuously roaming the area and reports any appearing task.

- The robotics network itself can do both, discovering and task allocation phases. In this case, robots have to be equipped with extra sensors for task discovery process, and this may be costly and inefficient. Because the robots will not utilize these sensors during the task execution.

- A gateway can be used to forward task advertisement messages to the robotic network. It is used as a bridge between a robotic network and an external system which is responsible for discovering the tasks (e.g. remote sensing system).

In the first three cases where the WSN, mobile sensors, and robotic network may have the capability to discover a task and estimate its resources and quality level requirements, in this case they can report the discovered task directly to the closest robot, which will be considered as an auctioneer. If they are not able to estimate the requirements of the discovered tasks then they may report the task to a central point in which task requirements are determined and then advertised to the robotic network via a gateway. We consider the general scenario in which the robot that receives the task advertisement message can be any robot and may or may not be the closest robot to the task.

3.6 The Proposed MOTA Objectives

In general, for MOTA problem, the effective solution has to assign each task to the most appropriate robot. In a market-based approach, the fitness function is used to express how optimal the assignment is. The greater the fitness value, the closer to the optimal solution [32]. The objectives we are trying to achieve in this approach are:

1. Minimize the total traveled distance to visit all assigned tasks by a robot R^i.

$$MIN(\sum_{j=1}^{k} ||p_R^i - p_T^j||) \ \forall \ T^j \in L^i \tag{3.4}$$

   where p_R^i, p_T^j are the positions of the robot and the task, respectively, and L^i is the list of length k for the assigned tasks.

2. Maximize quality satisfaction, in other words, minimize the difference between the robot quality q_R^i and the average quality of tasks executed by it R_q^i:

$$MINI(|q_R^i - R_q^i|) \tag{3.5}$$

$$where\ R_q^i = \frac{\sum_{j=1}^{l} q_T^j}{l} \tag{3.6}$$

where l is the number of tasks executed by a robot R^i.

3. Balance the load among robots, such that at any point of time for any two robots R^i, R^j minimize the difference between the number of assigned tasks for each e.i. $MIN(|L^i| - |L^j|)$. This local optimally criteria will lead to a global solution. However, it may lead to sub-optimal solution considering other objectives such as traveled distance, quality satisfaction.

4. Balance and minimize energy consumption.

5. Balance resource consumption.

Without loss of generality, we assume there are three robots $R = \{R^0, R^1, R^2\}$ are starting with the same amount of energy and resources, however with different levels of quality $q_q^2, q_q^5,\ and\ q_q^8$ respectively. Balancing resource and energy consumptions eventually lead to maximizing network lifetime which defines as the time elapsed before the first robot dies due to energy shortage or because it cannot accept new task due to the lack of energy or resources. To satisfy the aformentioned objectives, biding value has to consider the following points:

- To minimize total traveled distance, a task in a position p_T^i has to be assigned to a robot in a position p_R^j such that $||p_T^i - p_R^i||$ is the minimum for all robots positions.

- The optimal quality satisfaction is achieved when the robot performs tasks within ϵ of its quality level. Fig.3.1 demonstrates the quality satisfaction for different types of robots.

- Load balanced achieved when a task assigned to a robot with the minimum number of assigned tasks among the robots participate in the task assignment.

- Assigning a task to a robot has the maximum resources ensure resource balance. Similarly, assigning a task to a robot with the maximum energy ensure energy balance among the connected robots.

3.7 Tasks Synergy

Synergy is a term used to describe the relationship between tasks; tasks have a positive synergy if the total cost of executing both of them by one robot is less than if each is executed by a different robot. In this work, we consider spatial synergy. Then we can re-define synergy based on distance as following: if the total traveled distance for visiting two tasks using one robot is less than the total traveled distance if each task visited by different robots, then these tasks has a positive synergy. Obviously, the negative synergy is the opposite.

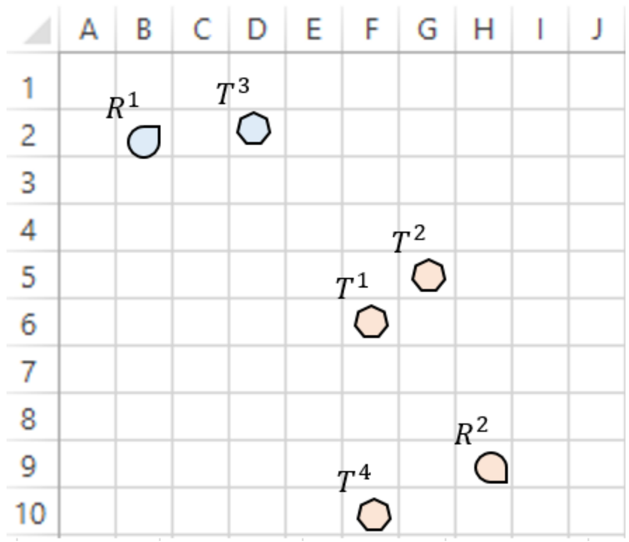


Figure 3.3: Synergy example: The optimal assignment is to assign T^3 to robot R^1 and T^4 to robot R^2 because they have negative synergy. While T^1 and T^2 have a positive synergy, they have to be executed by one robot R^2

For instance, in Fig.3.3 two robots R^1, R^2 bid for two tasks T^1, T^2. Tasks T^1, T^2 have a positive synergy; meaning the total cost of performing both tasks by one robot is less than the cost if each one is executed by different robots. In this example R^2 wins task T^1, then it bids for task T^2 as if it is on the location of task T^1 , hence R^2 wins T^2 as well. In contrast, tasks T^3, T^4 have a negative synergy i.e. the cost of executing both of them by one robot is higher than the cost if they are executed by different robots. Therefore, robot R^1 wins T^3 and R^2 wins T^4.

The proposed MOTA consider tasks spatial synergy. It ensures that if a robot wins a task T^k, then it is more likely to win other tasks that have a positive synergy with T^k. Therefore, for a task T^k in the current auction a robot R^i bids based on the minimum distance between task location p_T^k and robot location p_R^i or any of its unaccomplished tasks $\hat{L}^i$ location. Hence, the traveled distance cost which a robot will bid based on is given using the following equation:

$$d_{ij} = MIN(||p_R^i - p_T^k||, ||p_T^j - p_T^k||) \ \forall \ T^j \in \hat{L}^i \tag{3.7}$$

3.8 Task Execution Sequence

We assume the order of the tasks execution is not important as it is stated in the system model 3.3. When a robot has more than one unaccomplished task in its list it will need to determine the route to visit them. Finding the minimum route is a variation of the Traveling Salesman Problem (TSP) [21]. We used the nearest neighbor(NN) algorithm [33], in which a robot orders its tasks in descending order based on their distance to its location to its current location and then execute them one after another as shown in Fig.3.4.

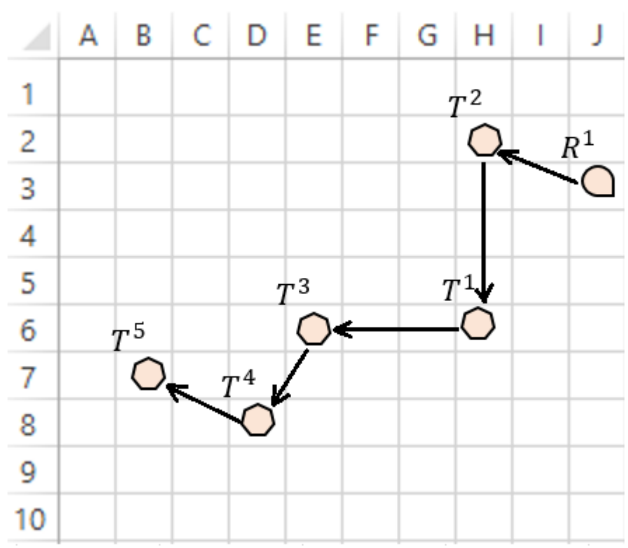


Figure 3.4: Robot R^1 visits assigned tasks using nearest neighbor algorithm.

Rather than performing the tasks in a First Come First Serve (FCFS) like fashion, apply the NN algorithm is considered as a second means of minimizing the traveled distance besides the including distance factor in the fitness function on Equation 3.10.

3.9 Multi-objective Auction-based Task allocation

The auction process is the main part of marked-based task allocation approach, robots cooperate with each other via an explicit negotiation. The proposed method uses a low-cost one-round single auction. Meaning, a task is assigned to a robot with the highest bid at one round. The auction implemented as contract net protocol (CNP), see section 1.3. The auction process consists of four sequential steps.

1. **Task advertisement:** The auction process starts once a robot discovers a new task or when it receives a new task advertisement message from an agent outside the system such as a remote sensing system in automation farming application. Task advertisement message includes all details of the task; its location, quality level, and the required resources to execute the task. Then the robot considers itself as an auctioneer for this particular task.

2. **Auction announcement:** Then, the auctioneer, which is the robot gets task advertisement, computes its bid for the task, as well as announces a new

auction by broadcasting an auction message (*auction_msg*) to its neighbors. The auction announcement message contains all details of the current tasks. The auction remains open for a sufficient set of time (*auction_time*) to allow bidders to send their bids.

3. **Bid submission:** Once a robot receives an auction announcement message, it computes its bid based on the proposed fitness function and submits it to the auctioneer. Unlike the conventional auction based bidding process, in which a robot computes its bid based on the distance only and assumes it has sufficient energy and resources to complete a task. The proposed approach considers the availability of energy and resources in the bidding process as well as quality satisfaction and workload balance. A bid is a computed scalar value using the fitness function and represents a robot's fitness for the task, the fitness function design explained in the next section.

4. **Close of auction and winner selection:** During the auction the auctioneer maintains a list ($BidsList$) for all current bids and associated bidder's ID, and once auction time ends the auctioneer selects the bidder with the maximum bid from the bids list $BidsList$ and assigns it the current task.

Fig.3.6 contrasts the state machine of a robot. A robot starts in an idle state and once it gets to know about new task it jumps to auctioneer state and runs Algorithm 1 in which it computes its bid and sends open auction message (auction_msg) to other robots. All robots which receive auction_msg turn from idle state to bidder state and run algorithm 2 and send their bids. The auctioneer then selects the winner bidder and send it task assignment message win_msg. After the task has been assigned all robots, auctioneer and bidders, turn to idle state again.

Auction process can take place anytime, and it happens in parallel with task execution process. In other words, if a robot is performing a task but not participating in an auction it is considered as being in an idle state from auction process perspective, Fig.3.6 shows the robot states.

Algorithm 1: Auctioneer algorithm

input : New task announcement
output: Assigning a task to the highest bidder

```
In parallel with other procedures (moving, executing task, etc.);
AuctioneerBid ← ComputeMyBid();
AppendToBidsList(AuctioneerId,AuctioneerBid) ;
BroadcastAuctionMsg();
AuctioneerTimer ← t sec ;
while !IsAuctionEnds() do
    if ReceiveBidMsg() then
        AppendToBidsList(bidValue,BidderID) ;
winnerBidder ← GetHighestBidder();
SendTaskAssignMsg(winnerBidder);
```

Algorithm 2: Bidder algorithm

input : Auction message, Task Assignment message
output: Bid for task, append assigned task to the task list

```
In parallel with other procedures;
if AuctionMsgReceived() then
    MyBid ← ComputeMyBid() ;
    sendBidMsg (MyBid,auctioneerID);
else if TaskAssignMsgReceived() then
    AppendToTaskList(NewTask);
```

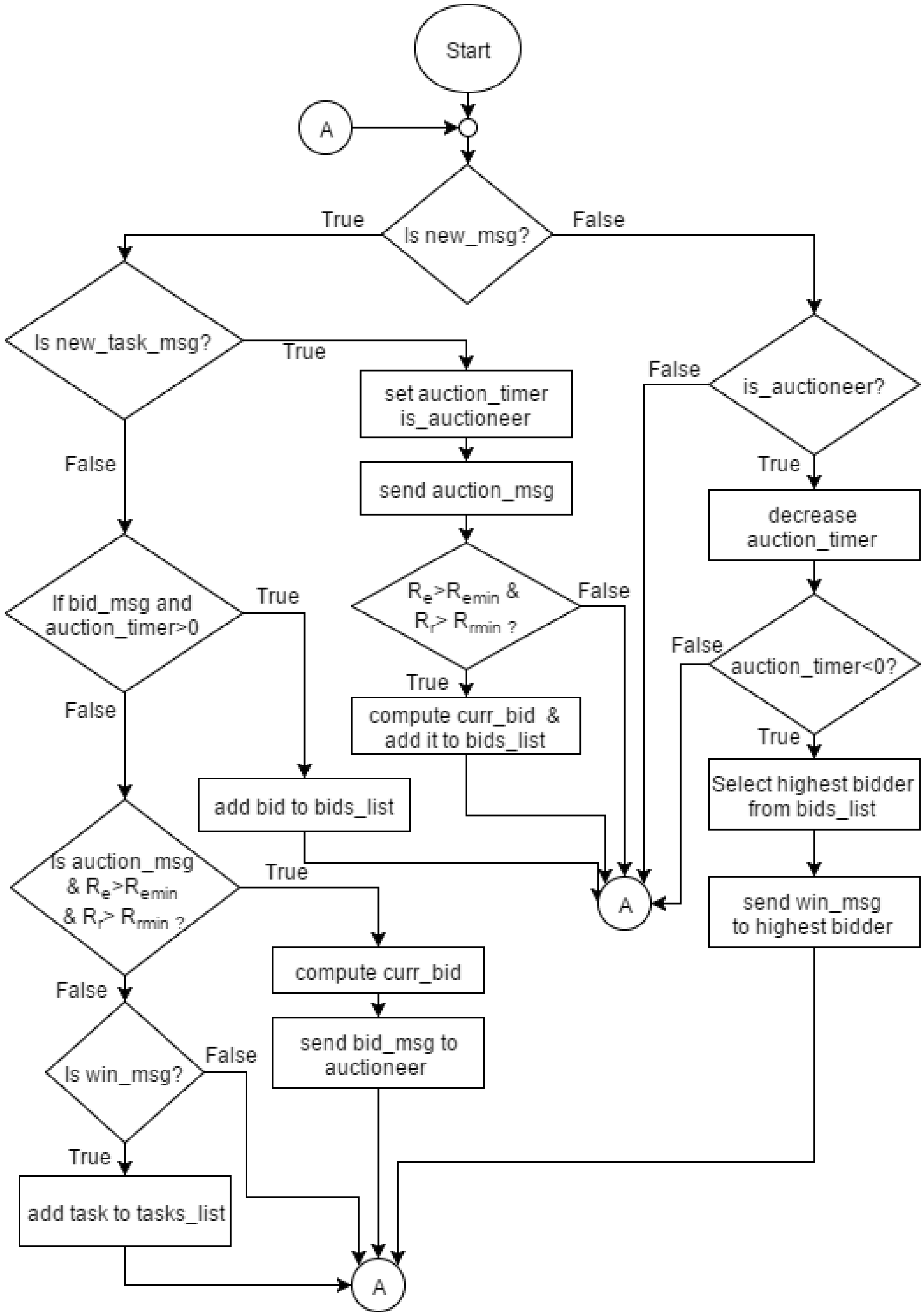


Figure 3.5: Auction process flow diagram

3.10 Limited Communication

Basically, we assume ideal communication between robots. However, in the case of limited communication a task may not get assinged to any robot if the task

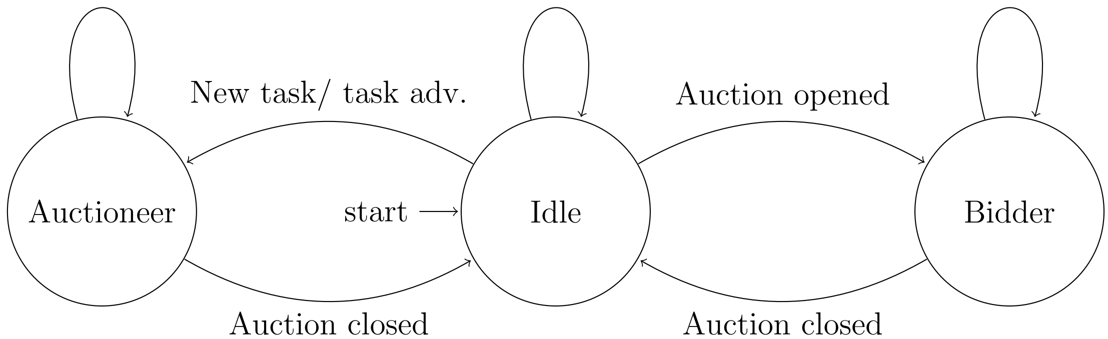


Figure 3.6: A robot state diagram in during auction process.

assignment message is lost. Therefore, we designed a three handshake task assignment protocol to prevent. Under limited communication, there is a possibility that the auctioneer selects the winner bidder and sends it winning message ($winMsg$) and it gets lost. In this case, from the auctioneer perspective, the task has been assigned while the winner robot knows nothing about the assignment. To minimize the occurrence of this problem we have proposed a three handshake assignment, as shown in Fig.3.7a. In the normal case, Fig.3.7a the auctioneer (A) sends $winMsg$, and the winner (W) replies by $acceptMsg$, and finally the auctioneer sends final *Ack* message to confirm that the assignment has been accepted. In case of Fig.3.7b, where $acceptMsg$ or $winMsg$ gets lost then the auctioneer will reassign the task to the next max bidder, the bidder, however, will ignore the task because it does not get the final *Ack* message from the auctioneer. The last case is shown in Fig.3.7c where the winner does not receive the *Ack* message, in this case only the auctioneer will assume that the task gets assigned while the winner ignores the task because it does not receive the final *Ack* message.

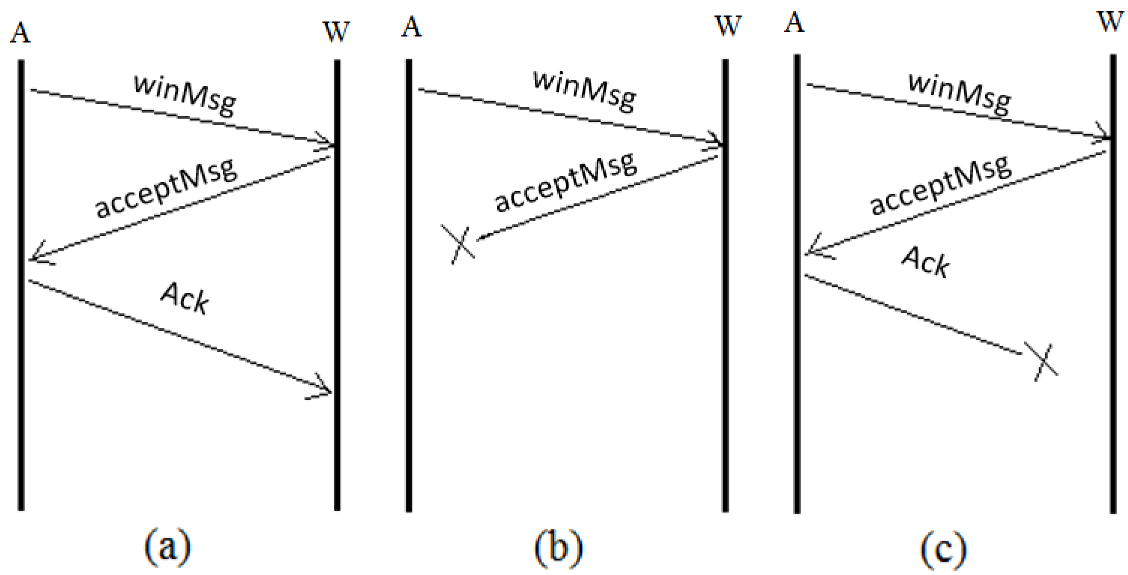


Figure 3.7: Three handshake task assignment between an Auctioneer A and a winner robot W.

3.10.1 Fitness Function

Robots bid with their fitness value that is computed by the fitness function. It is an objective function that tells how optimal the solution is. The fitness function has a great influence on the performance of the proposed approach. Thus, we design the fitness function to include all system objectives listed in section 3.6. Usually, the fitness function consists of one factor, the traveled distance between a robot and a task ([34], [35], [22], [26]). However, in our approach, we have multiple factors to deal with, consequently, there is a need for a multi-factor decision model. In this thesis we have utilized two well-known models for this: a) The weighted sum decision model [36] b) Fuzzy logic system [37]. Both models are used in the proposed auction-based approach to evaluate the fitness value in which a robot bids, and hence the auctioneer use in the selection process (assignment process).

The required information for computing the fitness value is available for all robots such that each task T^j is announced in the auction with its location p_T^j, required quality level q_T^j, and required resources τ_T^j. A robot R^i knows its status at current time t: its location p_R^i, quality level q_R^i, its available resources τ_R^i, and available energy E^j. Therefore, a robot R^i , in both state an auctioneer or a bidder, keep track of this information and used them in the auction process.

3.10.2 Factors Normalization

In order to combine different objective factors where each of which has its own range, they have to be normalized. Therefore, all factors that are stated in the proposed approach objectives in section 3.6 are normalized either over their maximum or average values.

- Distance $(d_{norm}) = \frac{d_{ij}}{D}$
- Difference in Quality $(q_{norm}) = \frac{|q_R^i - q_T^j|}{max(\Delta q)}$
- Load $(l_{norm}) = \frac{L_{max} - |L^i|}{L_{max}}$
- Energy $(E_{norm}) = \frac{E^i}{E_{max}}$
- Resources $(\tau_{norm}) = \frac{\tau_R^i - \tau_T^i}{\tau_{max}}$

Where: d_{ij}: The minimum distance between a task T^j and a robot R^i or any of its unaccomplished task, as it is computed in Equation 3.7. $D = |A|$: The diagonal of the area (the maximum distance between a task and a robot). $max(\Delta q)$: the maximum difference between a robot quality level and a task quality level. $L_{max}, |L^i|$: the maximum number of task per robot, and the number of tasks assigned to a robot, respectively. τ_{max}: The maximum resources available in a robot.

3.10.3 The Weighted Sum Decision Model (WSM)

The general fitness function using weighted sum model is shown in Equation 3.8.

$$F_{R^i,T^j} = -W_d * d_{norm} - W_q * q_{norm} + W_t * l_{norm} + W_e * E_{norm} + W_r * \tau_{norm} \tag{3.8}$$

Where: $W_d, W_q, W_t, W_e, W_r \in \mathbb{R}_{\geq 0}$ such that $W_d+W_q+W_t+W_e+W_r = 1$. The weights can be adjusted based on the application to achieve different objectives.

The general fitness function in Equation 3.8 is composed of five terms and it is designed to satisfy the objectives of the proposed MOTA. The first term represents the inverse relationship between the fitness function and the distance between the robot and task, i.e. $F_{R^i,T^j} \propto 1/d_{ij}$, the aim is to minimize the traveled distance. The second term reflects the task quality satisfaction where the difference between a task quality q_T^j and robot quality q_R^i needs to be minimized. Moreover, the third term ensures load balance among all robots. The fourth and fifth terms, consider the availability of energy and resources in order to count for assigning robots with sufficient energy and resources to execute the task.

Weights Selection

Auction-based task allocation approach heavily depends on the fitness function design, which determines whether a task will be assigned and to whom. The proposed general fitness function as in Equation 3.8 can be adjusted to be biased for one or more factors using its associated weights. For instance, to treat all factors equally all weights have to be equal to $W = 1/5 = 0.2$ and to consider distance factor only its associated weight has to be equal to one $W_d = 1$, consequently the rest of weights are zeros.

Obviously, considering distance factor only leads to insufficient quality satisfaction because there is no relation between the task quality and its location, each one is uniformly distributed among incoming tasks. Hence, distance and quality factors are independent factors.

A Monte Carlo experiment has been conducted using Matlab to determine the appropriate weights for distance and quality factors W_d,W_q, respectively. Fig.3.8 shows the relation between the traveled distance and the drift from quality satisfaction with respect to the change in W_d ($W_q = 1 - W_d$). It is clear that the best quality satisfaction is when $W_q = 1 (i.e. W_d = 0)$, which is the worst case for traveled distance. When W_d increases (W_q decreases) consequently the traveled distance decreases, and the shift from quality satisfaction increases. The point at which the two objective functions intersects (i.e. $W_d = 0.7$, $W_q = 0.3$) is the optimal setting for both factors. These weights will be used in the fitness function. These weights are valid for three quality levels for robots and three or more qual-

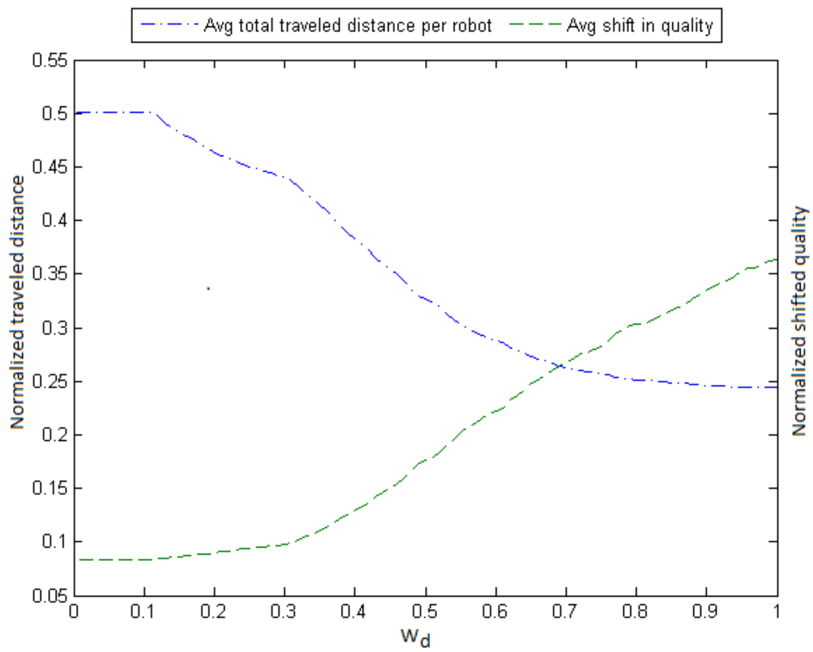


Figure 3.8: The effect of increasing the value of W_d on the traveled distance and the drift of quality. The increase of W_d decreases the traveled distance (blue line), however increases the drift of the quality. The best setting is when $W_d = 0.7, W_q = 1 - 0.7 = 0.3$.

ity levels for tasks. Fig.3.9 illustrates that changing the number of tasks quality levels has no effect on the optimal weights of distance and quality (W_d, W_q) as far as the relationship defined in Equation 3.1 is satisfied.

In order to investigate the relationship between these two factors and the other factors, we have simulated the proposed auction-based approach considering these two factors only. We observed that satisfying distance and quality objectives lead to minimize the total traveled distance and maximize the quality satisfaction. Moreover, it leads to balance the load and the energy and resource consumption among all robots. Because satisfying the task quality means that each robot executes the tasks that have quality level $\pm\alpha$, and because the quality of the tasks are

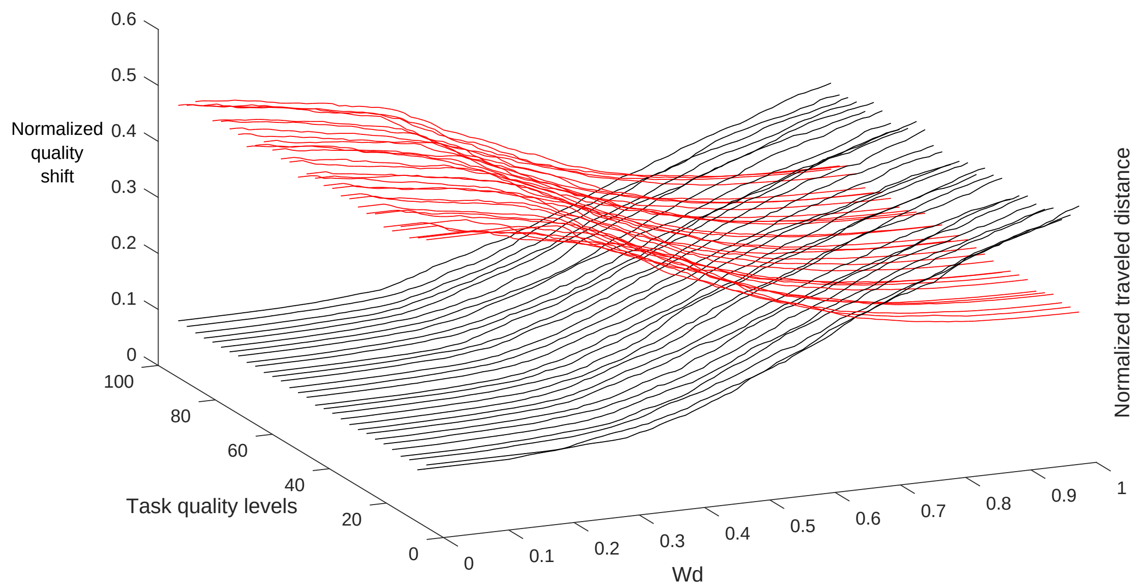


Figure 3.9: The effect of increasing the value of W_d on the traveled distance and the drift of quality, with respect to different number of tasks' quality. The best setting is when $W_d = 0.7, W_q = 1 - 0.7 = 0.3$.

uniformly distributed then each robot gets almost same number of tasks (load balancing). Now since workload balancing exist so does the resource consumptions, which is uniformly distributed among the tasks. Considering that movement is the main energy consumption source, and balancing the traveled distance is satisfied then energy consumption is also balanced. Table 3.1 shows the results of the experiment for 1000 runs for 1000 tasks. Table 3.1 results demonstrate that residual energy E, resources R, and a number of tasks executed per robot are almost the same for all robots with a tiny fraction of error when only distance and quality are considered ($W_d = 0.7$, $W_q = 0.3$).

The general fitness function is re-designed to be based on only the two factors;

Number of tasks = 1000, Number of iterations=1000			
	Residual energy	Residual resources	Task executed per robot
Mean	542.3	1832.7	333.33
Error (±)	0.81	1.8264	0

Table 3.1: The auction-based proposed method considering traveled distance and quality satisfaction objectives leads to balance energy, resources, and load among all robots with a relative small error.

distance and quality with $W_d = 0.7, W_q = 0.3$ as shown below :

$$F_{R^i,T^j} = -0.7 * \ d_{norm} - 0.3 * \ q_{norm} \tag{3.9}$$

Table 3.2 illustrates the impact of different weights settings on the proposed MOTA. The scenario includes three robots $R^1, R^2, and R^3$ with quality levels $q_R^1 = 2, q_R^2 = 5, q_R^3 = 8$ respectively. Three tasks (T^1, T^2, T^3) appear in random locations and quality levels. The distance between a task and a robot is denoted by d_{ij} and (T^j, q_T^j) represents the task number and its quality levels. If robots bid based on distance only $W_d = 1$, robot R^1 wins task T^1, robot R^2 wins tasks T^2, T^3, this yields the minimum traveled distance. However, it does not satisfy quality level objective e.g. robot R^1 with quality level $q_R^1 = 2$ wins task T^1 with quality level, $q_T^1 = 9$. In contrast, when robots bid based on quality factor only i.e. $W_q = 1$, the quality satisfaction gets high but robots may travel longer distance such as robot R^3 with task T^3. In the last part of Table 3.2, robots calculate their fitness based on $W_d = 0.7, W_q = 0.3$. It leads to balance between the two objectives; minimizing the traveled distance, like the case of robot R^2 with task T^3, as well as maximizing the quality satisfaction, like the case of robot R^1 and task T^2.

	($T^1, q_T^1 = 9$)		($T^2, q_T^2 = 1$)		($T^3, q_T^3 = 8$)	
	$W_d = 1, W_q = 0$					
(R^i, q_R^i)	d_{ij}	Bid	d_{ij}	Bid	d_{ij}	Bid
$(1, 2)$	0.25	-0.25	0.38	-0.38	0.50	-0.50
$(2, 5)$	0.63	-0.63	0.20	-0.20	0.30	-0.30
$(3, 8)$	0.38	-0.38	0.50	-0.50	0.38	-0.38
	$W_q = 1, W_d = 0$					
(R^i, q_R^i)	d_{ij}	Bid	d_{ij}	Bid	d_{ij}	Bid
$(1, 2)$	0.25	-0.87	0.25	-0.12	0.38	-0.75
$(2, 5)$	0.63	-0.5	0.38	-0.50	0.03	-0.37
$(3, 8)$	0.38	-0.12	0.50	-0.87	0.63	0.00
	$W_d = 0.7, W_q = 0.3$					
(R^i, q_R^i)	d_{ij}	Bid	d_{ij}	Bid	d_{ij}	Bid
$(1, 2)$	0.25	-2.27	0.13	-0.38	0.38	-2.06
$(2, 5)$	0.63	-1.63	0.25	-1.37	0.03	-0.91
$(3, 8)$	0.38	-0.56	0.50	-2.45	0.63	-0.43

Table 3.2: Auction based on various weight settings example. First raw includes tasks and their quality level, first column includes robots and their quality level. Dark gray cells represent an assignment between a robot in the raw and a task on the column.

Synergy Side Effect

Although synergy reduces the total cost, it has its side effect on the workload balance objective in the proposed MOTA. In Fig.3.10 robot R^1 won all the tasks , while robot R^2 won nothing (e.i. using Equation 3.7 $d_{1x} < d_{2x} \forall x = 1, 2, 3, 4$). We call this problem jump to the middle problem (JMP). It comes because once the robot R^1 wins a task that locates close to or in the middle of the are of interest, it gets close to most of all new tasks that are going to appear in future, either directly or via its unaccomplished tasks, and consequently wins more and more tasks, leading to compromised load balancing objective. Therefore, we had to re-design the fitness function in Equation 3.9 to include the workload balance objective of the proposed general fitness function in Equation 3.8. Since, we do

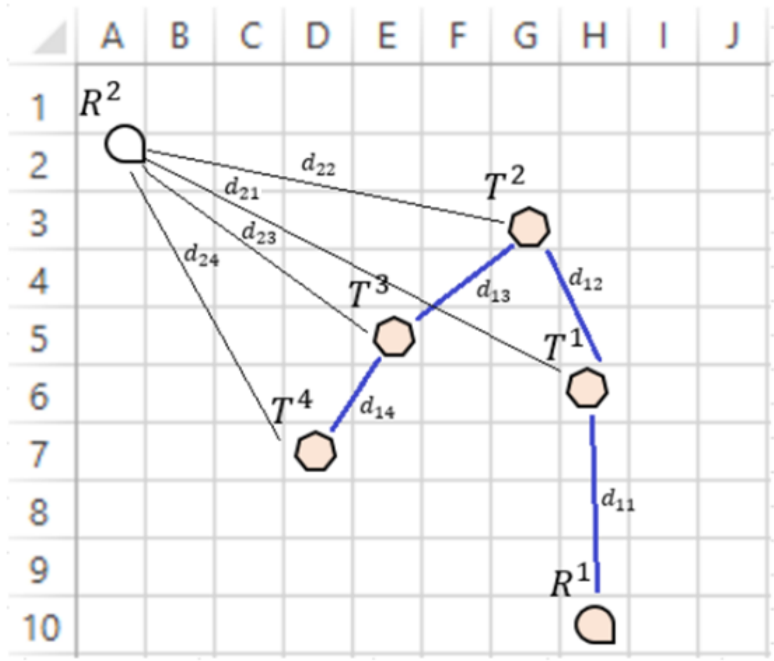


Figure 3.10: Synergy side-effect: R^1 jumps to the middle of the are, hence it occupied all tasks.

not prefer one objective over the others, we set each associated weight to 1/3, i.e. $W_d = W_q = W_t = 1/3$. Then, we keep the weight of $W_t = 1/3$, and the remaining 2/3 divided between W_d, W_q proportionally based on the results from Fig.3.8, $W_d = 0.7 * 2/3 = 0.47, W_q = 0.3 * 2/3 = 0.23$. Hence, the final wighted sum fitness function which will be used upword is re-designed as follows:

$$F_{R^i,T^j} = -0.46 * \ d_{norm} - 0.21 * \ q_{norm} + 0.33 * \ l_{norm} \tag{3.10}$$

3.10.4 Fitness Based on Fuzzy Logic System

Fuzzy logic has been introduced in 1965 by the mathematician Lotfi Zadeh [38] as an alternative for classical boolean logic to deal with the uncertainty. It consists of three main processes: a) Fuzzification inwhich crisp input variables are converted into fuzzy membership function, b) Inferencing where the applicable rules are executed, c) Defuzzification where the output variable is produced as a crisp value.

We have also utilized fuzzy logic to combine the objectives we have stated in section 3.6 and produce a fitness value that represents the suitability of a robot to perform a task. In contrast to the WSM, where the weights determine the behavior of the system, in fuzzy logic the rules and membership functions are used instead. The input variables of the fitness fuzzy system are the normalized factors included in Equation 3.10, and the output variable is the fitness value, as shown in Fig.3.11a. The proposed fuzzy system solution attempts to minimize the total traveled distance, distribute the load equally among the robots, and minimize the difference between a quality level required by a task and assigned robot quality (maximize the quality satisfaction). This have been implemented by the input and output membership functions which are designed as follows:

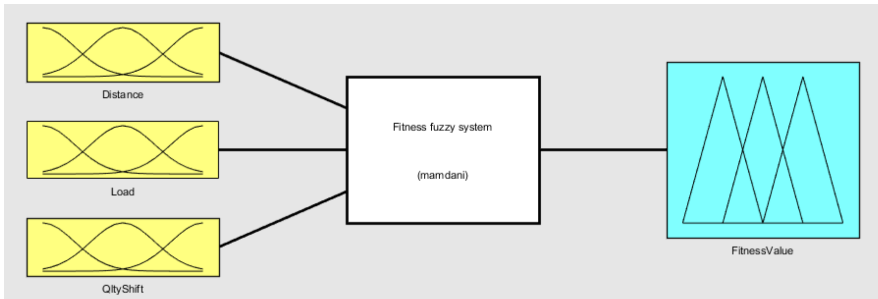
a) The distance and load input variables are formed as three triangular-shaped membership functions Fig.3.11b, which is a typical function for a case where as the input variable increases its backwards membership decreases (e.g. low membership function), and frontwards membership function increases (e.g. medium membership function), which is the case for distance, load. The triangular-shaped membership functions are customized using a, b and c variables which are driven from the number of robots n as it shown in first part of

Table 3.3.

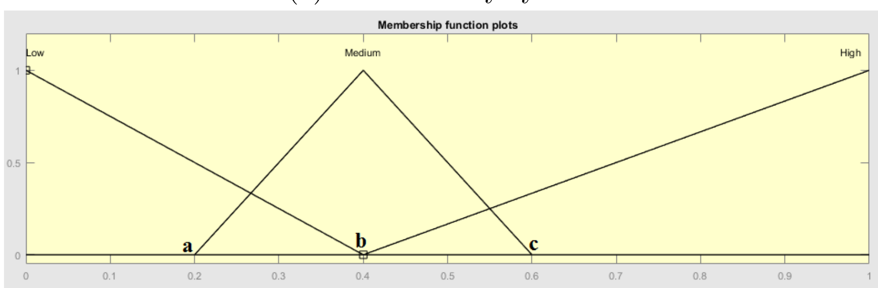
b) The quality input variable consists of two trapezoidal-shaped membership function for low and high, and one triangular-shaped for medium. We used trapezoidal-shaped for low and high membership functions because there is a clear part where the difference is 100% low (i.e. [0,a]) or high (i.e. [d,1]). While the medium membership function is a transportation phase between low and high membership functions, hence, it is presented as triangular function. Note that the variables a, b, c, and d, in the middle section of Table 3.3, shape quality membership functions. These variables are directly driven from the α variable (Equation 3.3) which determines the range of a robot quality. For example if quality shift is less than α it considers as low because it is in the range of a robot quality.

c) The fitness value (output variable) includes four triangular-shaped membership functions for low, medium, high and very high fitness value, see Fig.3.11d.

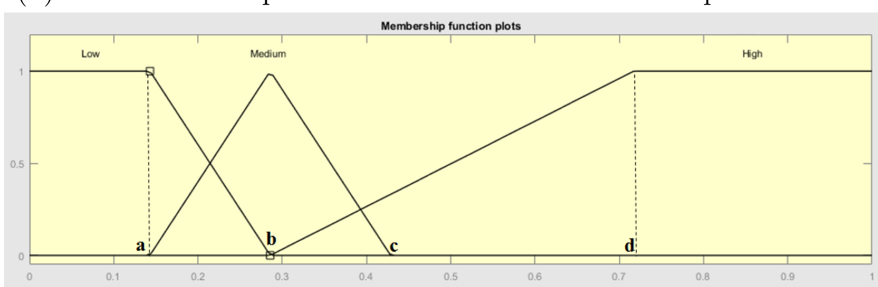
Generally, there is no way to tell what is the optimal number of membership functions for a certain input/output fuzzy variable. We choose three functions because this is the case for most scenarios where you have low, medium and high values. However, for the output variable we make it four membership functions because there is a need to differentiate between the permutation of low,low, low/medium and low, low, high for the input variables. Therefore, as Table 3.4 shows the inference fitness value rule for the first case is $Very\ high$ and for the second one is $High$.



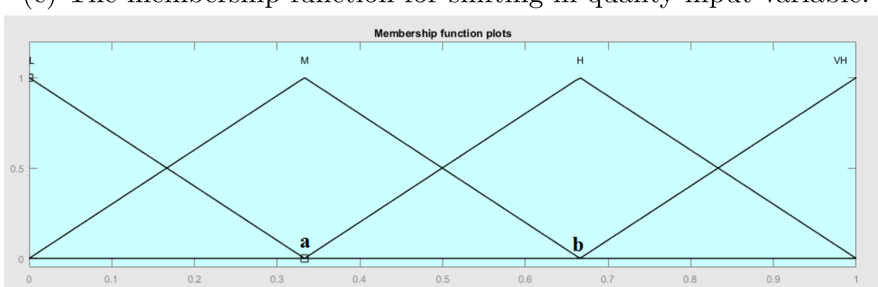
(a) Fitness fuzzy system.



(b) The membership function for distance and load input variables.



(c) The membership function for shifting in quality input variable.



(d) Fitness value (output variable).

Figure 3.11: Fitness fuzzy system for task allocation.

Distance, load membership functions	
Low	$m_L(x) = \begin{cases} \frac{b-x}{b} & \text{if } 0 \leq x \leq b \\ 0 & \text{if } x > b \end{cases}$
Medium	$m_M(x) = \begin{cases} 0 & \text{if } x < a \text{ or } x > c \\ \frac{x-a}{b-a} & \text{if } a \leq x \leq b \\ \frac{c-x}{c-b} & \text{if } b < x \leq c \end{cases}$
High	$m_H(x) = \begin{cases} 0 & \text{if } x < b \\ \frac{x-b}{1-b} & \text{if } b \leq x \leq 1 \end{cases}$
	where $a = \frac{1}{3\sqrt{n}}, b = 2a, and\ c = 3a.$
Quality difference membership functions	
Low	$m_L(QS) = \begin{cases} 1 & \text{if } QS \leq a \\ \frac{b-QS}{b-a} & \text{if } a < QS \leq b \\ 0 & \text{if } QS > b \end{cases}$
Medium	$m_M(QS) = \begin{cases} 0 & \text{if } QS < a \text{ or } QS > c \\ \frac{QS-a}{b-a} & \text{if } a \leq QS \leq b \\ \frac{c-QS}{c-b} & \text{if } b < QS \leq c \end{cases}$
High	$m_H(QS) = \begin{cases} 0 & \text{if } QS < b \\ \frac{x-b}{d-b} & \text{if } b \leq QS \leq d \\ 1 & \text{if } x > d \end{cases}$
	where $a = \alpha/2, b = \alpha, c = 2\alpha, and\ d = 5\frac{\alpha}{2}.$
Fitness value membership functions	
Low	$m_L(x) = \begin{cases} \frac{a-x}{a} & \text{if } x \leq a \\ 0 & \text{if } x > a \end{cases}$
Medium	$m_M(x) = \begin{cases} \frac{x}{a} & \text{if } 0 \leq x \leq a \\ \frac{b-x}{b-a} & \text{if } a < QS \leq b \\ 0 & \text{if } x > b \end{cases}$
High	$m_H(x) = \begin{cases} 0 & \text{if } x < a \\ \frac{x-a}{b-a} & \text{if } a \leq x \leq b \\ \frac{x-1}{1-b} & \text{if } b < x \leq 1 \end{cases}$
Very high	$m_{VH}(x) = \begin{cases} \frac{x-b}{1-b} & \text{if } x \geq b \\ 0 & \text{if } x < b \end{cases}$
	where $a = \frac{1}{3}, b = \frac{2}{3}.$

Table 3.3: Fuzzy system membership functions.

Distance	Load	Quality difference	Fitness value
Low	Low	Low	Very high
Low	Low	Medium	Very high
Low	Low	High	Medium
Low	Medium	Low	Very high
Low	Medium	Medium	Medium
Low	Medium	High	Medium
Low	High	Low	High
Low	High	Medium	Medium
Low	High	High	Low
Medium	Low	Low	Very high
Medium	Low	Medium	High
Medium	Low	High	Low
Medium	Medium	Low	High
Medium	Medium	Medium	Medium
Medium	Medium	High	Low
Medium	High	Low	Medium
Medium	High	Medium	Medium
Medium	High	High	Low
High	Low	Low	High
High	Low	Medium	Medium
High	Low	High	Low
High	Medium	Low	Medium
High	Medium	Medium	Medium
High	Medium	High	Low
High	High	Low	Low
High	High	Medium	Low
High	High	High	Low

Table 3.4: Fuzzy rule base for fitness fuzzy system.

3.11 Threshold-based Multi-Objective Task Allocation

In the threshold-based method, a robot chooses to execute a task based on its response threshold and a stimulus for a task. Therefore, there is no explicit communication among robots for allocating the tasks. We have modified the threshold method to deal with multi-threshold values instead of single threshold value such that a robot will not participate in a task unless its stimuli are all greater than a robot's thresholds, as well as we added a communication routine to solve the tie problem. Each robot maintains the following thresholds:

Distance threshold It has been set such that it makes the area of interest is covered by all robots, and it is given by the following equation:

$$D_{th} = \frac{|A|}{2\sqrt{n}} \tag{3.11}$$

where n is the number of robots and $|A|$ is the diagonal of the area of interest. Fig.3.12 illustrates how D_{th} looks like for 9 number of robots.

Quality shifting threshold Meaning the maximum allowed shift between robot's quality and task's quality, and it is illustrated in the problem formulation in section 3.4. It has been set such that each robot will cover a cluster of the tasks' quality levels. Therefore, quality shift threshold set to keep each robot in its range and it is calculated using the following equation:

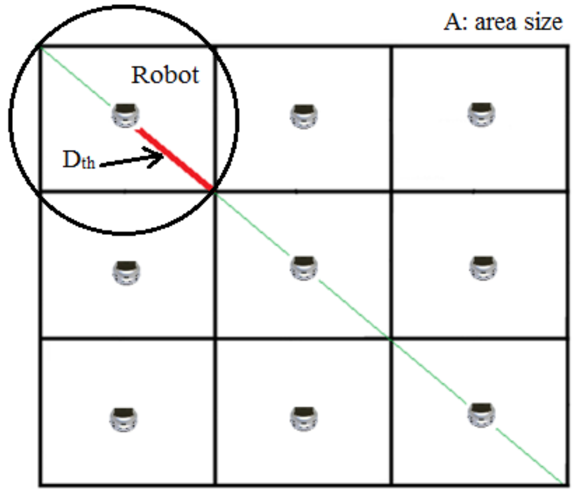


Figure 3.12: Illustration for D_{th} for 9 robots.

$$Q_{th} = \frac{\alpha}{2} \tag{3.12}$$

Where α is computed using Equation 3.1.

Load threshold It is the maximum number of tasks a robot can execute (i.e. maximum load); $L_{th} = L_{max}$

Task assignment using the proposed threshold-based approach has three cases:

1. Assigned to a single robot.
2. Assigned to more than one robot (Tie problem).
3. Did not get assigned.

The fist case is the desired case. However, the tie problem in the second case has been solved using an explicit communication, such that any robot broadcasts a tie avoidance message (which includes its ID and task number) before it attempts to execute the task. Then the robot with the maximum ID wins the task and the others will ignore it. In case three, when a task does not get assigned in the first iteration, it has to be announced again until it gets assigned. In each announcement, robots increases their thresholds (D_{th}, Q_{th}) by values related to the ratio between their current load $(|L|)$ and the maximum load L_{max}. The increase in quality and distance thresholds are given by the following equations:

$$Q_{inc}(t) = \begin{cases} Q_{th}(i-1) + Q_{min} & if\ |L_i| \leq L_{max}, Q_{th} < Q_{max} \\ 0 & Otherwise \end{cases}$$

Where Q_{min} is the increasing value and we set it as the minimum quality level. Basically it is a decision maker choice who can set it based on the number of quality level of tasks.

$$D_{inc}(t) = \begin{cases} D_{th}(i-1) * \alpha(1 - \frac{|L_i|}{L_{max}}) & if\ |L_i| \leq L_{max}\ and\ D_{th} < \frac{R_{com}}{2} \\ 0 & Otherwise \end{cases}$$

We set $\delta = 0.5$ to decrease the number of iteration required for D_{th} to occupy a task (i.e. to satisfy the threshold condition $d_{ij} < D_{th}$).

Lemma 3.1 *Given one robot in an area A, a robot distance threshold D_{th} is going to occupay any task in the area by maximum i iteration where i is given by the* *following equation:*

$$i = \frac{\log 2}{\log(1+\delta)}$$

To proof lemma 3.1, let's assume the extreme case where a robot locates on the corner of the square area and the task is on the opposite coroner then:

$$D_{th} = \frac{|A|}{2 * \sqrt{n}} = \frac{\sqrt{2}}{\sqrt{1}} = \frac{1}{\sqrt{2}}$$

Given that $D_{th}(i) = D_{th}(i-1) + D_{inc}$, then the robot will occupy the task when $D_{th} = |A|$

$$\sqrt{2} = \frac{1}{\sqrt{2}}(1-\delta)^i \implies i = \frac{\log 2}{\log(1+\delta)}$$

Fig.3.13 shows that the maximum δ the minimum number of iteration required to satisfy the distance threshold value.

Lemma 3.2 *Given two robots with distance threshold D_{th}, and communication range R_{com} they will be in the communication range of each other if $D_{th} < \frac{R_{com}}{2}$.*

Using lemma 3.2 robots will be able to solve the tie problem as long as they keep D_{th} less than half of their communication range.

3.12 Conclusion

In this chapter, we introduce a quality level as a new factor to be considered in the task allocation process. We also proposed a novel method for considering tasks synergy in dynamic scenarios. Also two distributed multi-objective task allocation

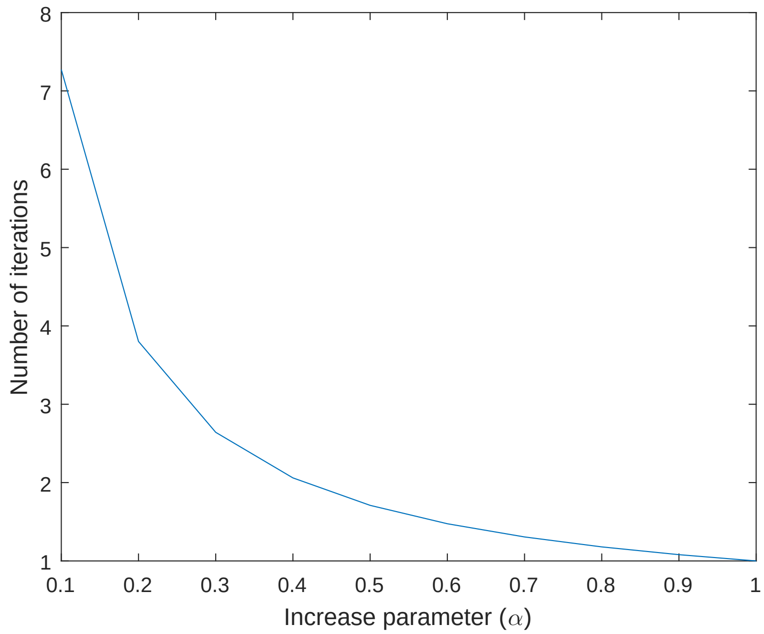


Figure 3.13: The relation between increment value δ and maximum number of iterations to occupy the tasks by D_{th} .

have been proposed for dynamic scenarios. The first approach based on auction and it comes with two flavors for combining the contradicted multi-objective a) weighted sum model, b) fuzzy logic system. The second approach is threshold-based and in which we extend the original single threshold approach to deal with multi-threshold.

CHAPTER 4

SIMULATION SETUP

We have validated our approach by evaluating its performance using KheperaIII robots on the Webots simulator [39] (Version 8.3.0). Webots is a fully-integrated design and coding platform, allowing for both virtual robot and environment design, it has prototypes for common real robots such as KheperaIII, E-puck, Pioneer, etc.

We have considered KheperaIII robot [40] as being well known and commonly used in such experiments, it is a differential wheeled robot with a dimension of 13cm diameter and 7cm height, with a ring of nine infra-red (IR) distance sensors which are used to detect obstacles, Fig.4.2 shows the real robot and the virtual robot in Webots simulator. The controllers of the robots are written in c programming language. The controller includes our proposed approach (MOTA) as well as the proportional-integral-derivative (PID) controller [41] which controls a robot navigation. The PID is a feedback control system which continuously calculates the difference between the desired goal and a measured current state.

In our case, a robot uses the PID to navigate to a task location (desired goal), and always computes the error (the difference between the current robot location and the task location) and minimizes it. The PID uses the odometry to estimate the current robot location, and utilizes IR sensors ring to avoid obstacles while driving a robot towards a task location.

4.1 Initial deployment

We have deployed three kheperaIII robots with energy E_{max}, resources τ_{max} enough to execute more than the expected number of tasks a robot may execute, and they have been assigned with quality levels: low, medium, and high. For the purpose of simulation, we will represents each quality level by a number (i.e. 2 for low, 5 for medium, and 8 for high). The tasks appear with higher number of quality levels than robots, we represent these levels from very low to very high quality level requirement with numbers from 1 to 9. Robots and tasks are given a unique color based on their quality level as following: Gray color for the low quality level , blue for the medium quality level, and red color for the high quality level. Fig. 4.1 illustrates the schematic diagram of the network for the simulation experiments. The robots form a robotics network and they are responsible for receiving tasks from an external agent and then allocating those tasks among themselves. As the robots in the robotics network work on the allocated tasks, they report their status (traveled distance, quality satisfaction, load, energy and resource consumption) to the external agent which can then be used to generate the simulation results. A full example of simulation experiment is shown in Fig.4.3.

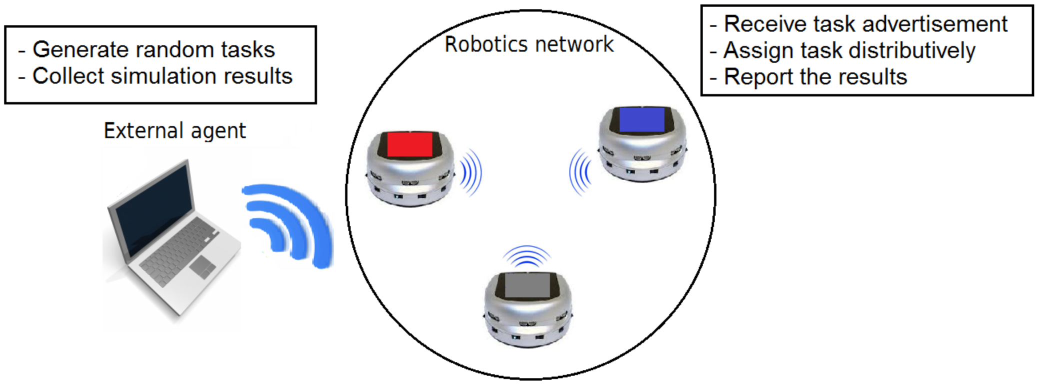
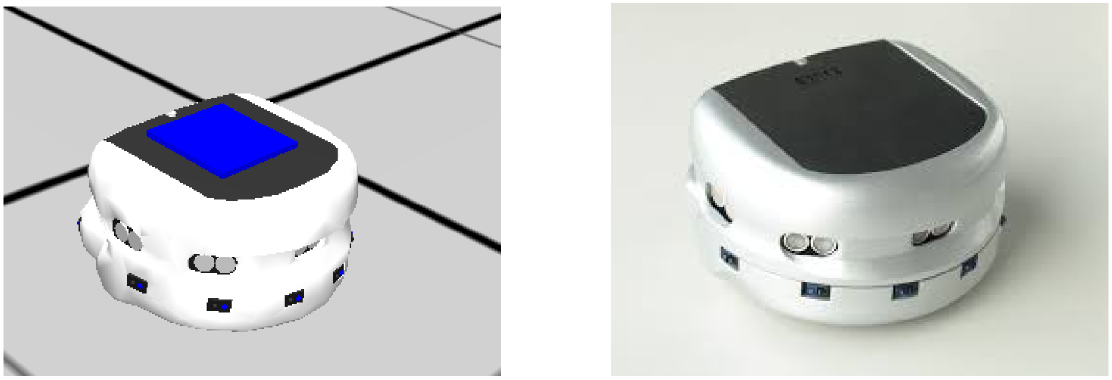


Figure 4.1: Network schematic diagram of the simulation experiments.

Parameter	Value
Number of robots	3, 6, 9, 12
Number of tasks	(12, 24, 36, 48)
Robot's quality (q_R)	(2, 5, 8)
Task's required quality (q_T)	(1, 2, ,3 , 4, 5, 6, 8, 9)
Area size	50m x 50m
Task inter-arrival mean time ($1/\lambda$)	80 sec
Task execution mean time ($1/\mu$)	140 sec
Task's required resources	(1, 2, ,3 , 4, 5, 6, 8, 9, 10)
PID parameters K_p, K_i, K_d	(2, 0.01, 0.04)
W_d, W_q, W_t	0.46, 0.21, 0.33
Number of Iterations	35

Table 4.1: Simulation Parameters.

Table 4.1 shows experiments parameters. The simulation starts by deploying the robots uniformly within the area. Then, tasks are generated following exponential inter-arrival time, and uniform location, resources, and quality requirements. A task advertisement message is produced with each newly generated task in the area. A random robot receives the task advertisement message and considers itself as an auctioneer for that task, and then the auction process (see



(a) KheperaIII within Webots (The blue plat on the top of the robot has been added to determine robot's quality

(b) A typical two drive wheels KheperaIII robot with 5 ultrasound sensors, 8 IR sensors, Wifi and bluetooth adapter.

Figure 4.2: KheperaIII robot

Section 3.9) takes place. Task advertisement message is sent by an external agent which is located outside the team of robots. It includes the details of the task; its location, resource, and quality requirement.

We have tested the proposed approach with a different number of tasks (12, 24, 36 and 48 tasks). In order to achive 95% confidence level, we have repeated each experiment 35 times. The confidence level of 95% is used to presents the collected results from each experiment. Video footage of an experiment of nine tasks is available at https://www.youtube.com/watch?v=HURvQlslTO8.

4.2 Conclusion

In this chapter, a fully integrated Webots simulator has been selected as the simulation tool for all simulation experiments and KheperaIII robot as a robot platform. We also state the initial deployment layout and parameter values for the simulation. A network schematic diagram shows the layout of the simulation

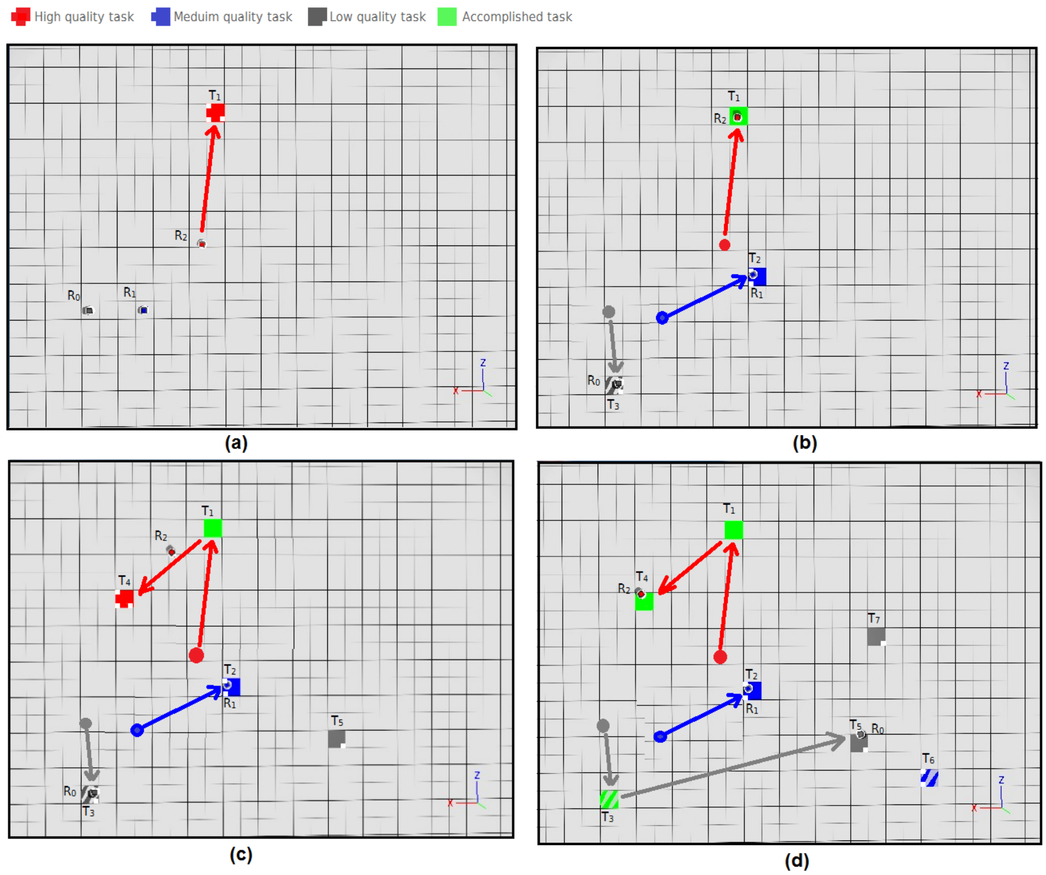


Figure 4.3: Proposed approach in Webots, a) Initial random deployment for three robots: low quality robot R_0, medium quality robot R_1, high quality robot R_2. New task T_1 appears and assigned to robot R_2, b) New tasks T^2 and Task T^3 assigned to robots R^1, R^0 respectively, and R^2 accomplished task T_1, c) Tasks T_4, T_5 appear and assigned to robots R_2, R_0 respectively. Since, R_2 is free it moves towards T_4, while R_0 is still busy with its previous task (T_3). d) More tasks emerge T_6, T_7 and assigned to R_0, R_1 respectively.

setup and the method used to collect data from the experiments.

CHAPTER 5

PERFORMANCE EVALUATION

In order to measure the performance of the proposed methods, we have tested them in different scenarios using Webots. Besides the performance results we got from Webots, we have also validated the proposed methods in a real experiment using Turtlebot2 robotics network. The comparison has been conducted between the different flavors of fitness computation (i.e. based on fuzzy logic and based on weighted sum model) as well as with the default auction-based approach, where a robot bids based on distance factor only. Moreover, we have formulated the problem mathematically and compared our results with simulation experiments. The list of performance metrics that have been used for evaluating the proposed methods are:

1. Total traveled distance per robot: which is the sum of traveled distance for a robot to accomplish all assigned tasks. Traveled distance for a robot

R^i which is assigned k number of tasks is computed using the following equation:

$$D^i_{total} = \sum_{j=1}^{k} d_{ij} \tag{5.1}$$

where d_{ij} is the distance between the robot R^i location and the nearest task T^j, and it computed using the Equation 3.7 as follows.

$$d_{ij} = MIN(||p^i_R - p^k_T||, ||p^j_T - p^k_T||) \; \forall \; T^j \in \hat{L}^i$$

2. Quality satisfaction: It measures the quality satisfaction of the tasks. Such that if the tasks have been assigned to robots with similar quality, then the average quality level for all assigned tasks for each individual robot is going to be almost equal to the robot's quality level, and it is computed as follows.

$$R^i_q = \frac{\sum_{j=1}^{l} q^j_T}{l}$$

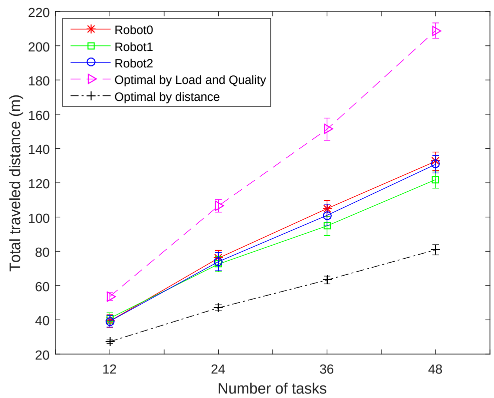
3. Load balance: It shows if the total load (the number of tasks) has been divided equally among the available robots or not. The optimal load balanceing is when all robots execute the same number of tasks. In another words, the the total number of tasks is divided equallly among all available robots.

Throughout this work, error bar represents a confidence interval with a confidence level of $\alpha = 0.05$.

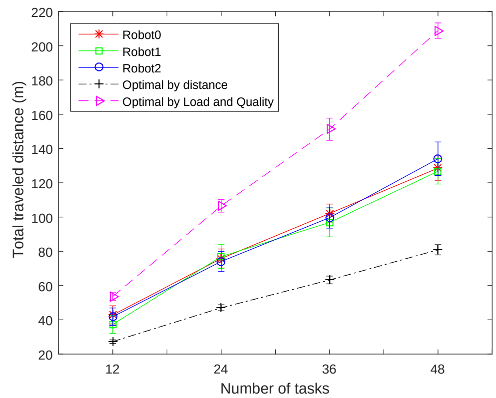
5.1 Simulation results

5.1.1 Traveled distance

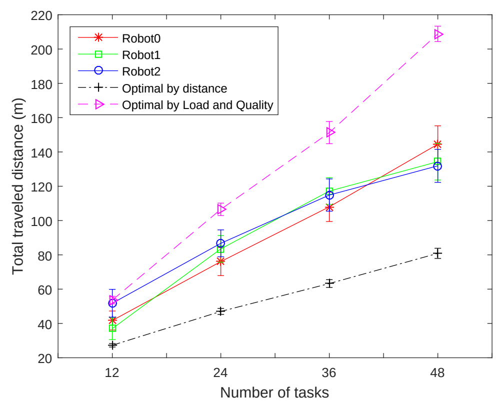
The first objective is to minimize the total distance traveled per robot, this objective has a significant weight in the fitness function ($W_d = 0.46$) in the auction-based approach. Besides that the distance factor has the highest weight, robots attempt to select the minimum route to visit/execute all allsigned tasks, which further decreases the total traveled distance. The fuzzy logic approach, on the other hand, gives almost same results, with a bit improvement in the variance especially when the number of tasks increases. Threshold-based produced relatively the worst results due to its nature of using the minimal cooperation. Fig. 5.1 shows the traveled distance per robot; although there is an increment in the total traveled distance per robot for the proposed multi-objective approaches compared to auction based on distance. This increase is expected because of the quality and load factors, which may lead to assigning a task to a further robot. Table 5.1 shows a comparison between the three proposed methods, it compares them based on the increase percentage in distance from the lower bound, and the upper bound. It shows that the weighted sum model (WSM) has the minimum increase in distance from the lower bound compare to other two methods (45.98 , 57.8, 58.48, 58.77), and it demonstrates the maximum drop compared to the upper bound (25.77, 30.36, 33.7, 38.5). Whereas the other two methods are almost comparable.



(a) Using weighted sum model



(b) Using fuzzy logic rules



(c) Using threshold-based approach

Figure 5.1: Total travelled distance per robot with varying number of tasks (area size =50mx50m)

The proposed synergy, see section 3.7, has a positive effect on minimizing the total traveled distance as shown in Fig.5.3. Moreover, when the number of tasks increases, the gain of using proposed synergy method to decrease the total traveled distance becomes more significant. That is the more number of tasks, the more synergy exists, and consequently the synergy method reduces a considerable distance. For example, the decrement in total traveled distance is (2.0%, 10.51%, 14.1%, 16.5%) when number of tasks is (12, 24, 36, and 48) respectively.

Number of Tasks	12	24	36	48
Lower Bound (LB)	27.2756	47.0080	63.2642	80.8730
Upper Bound (UB)	53.6379	106.5118	151.2789	208.8499
Using WSM	39.8163	74.1772	100.2636	128.3988
(+%) from LB	45.9777	57.7970	58.4839	58.7660
(-%) from UB	25.7683	30.3578	33.7227	38.5210
Using Fuzzy	43.5023	82.0874	113.3471	136.9086
(+%) from LB	59.4916	74.6243	79.1647	69.2884
(-%) from UB	18.8963	22.9312	25.0741	34.4464
Using Threshold	40.6833	75.5991	99.5364	129.7603
(+%) from LB	49.1564	60.8218	57.3345	60.4495
(-%) from UB	24.1520	29.0228	34.2034	37.8691

Table 5.1: Total traveled distance comparison between the three proposed methods with respect to the lower and upper bound

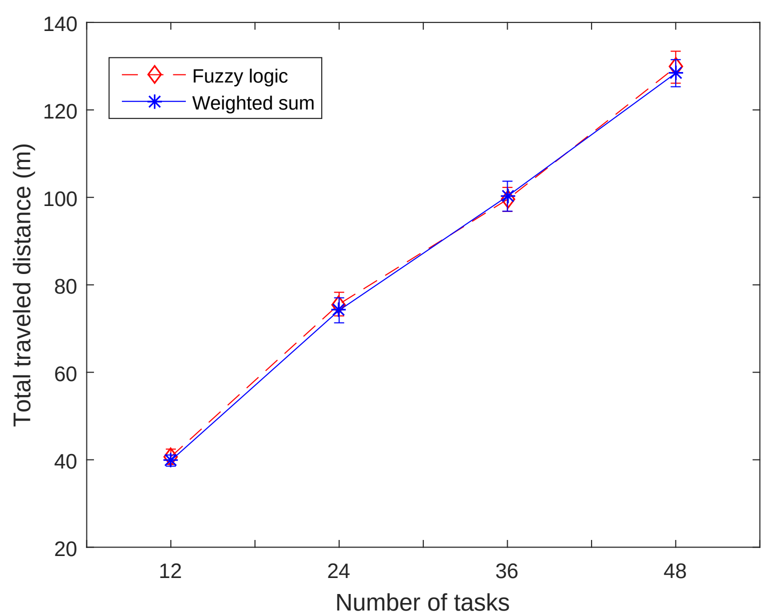


Figure 5.2: Compare average total traveled distance per robot in each of fuzzy logic and weighted sum fitness estimation (area size =50mx50m)

5.1.2 Quality satisfaction

In terms of quality satisfaction, the proposed approaches have also satisfied quality with a small deviation as shown in Fig. 5.4 where almost each robot performs tasks with the same or close quality to its quality requirement.

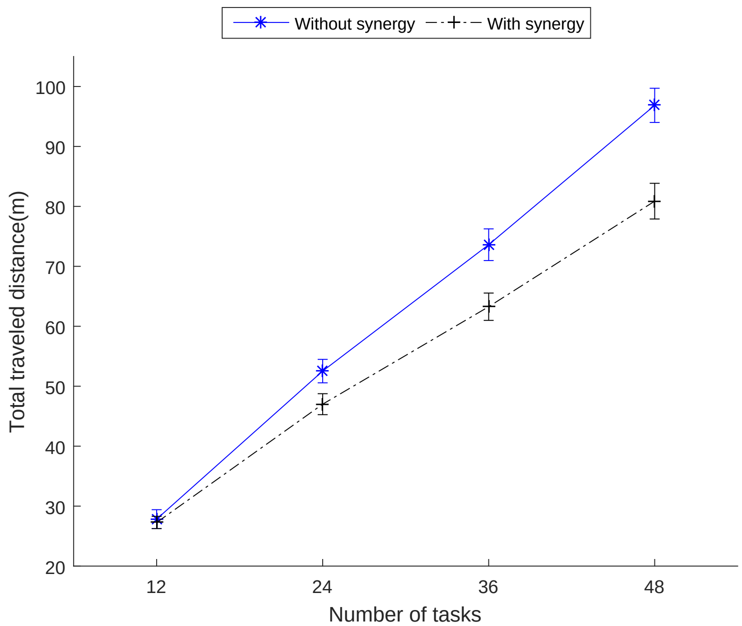


Figure 5.3: Average traveled distance with and without proposed synergy method if distance only considered in task assignment.

Number of Tasks	12	24	36	48
Without Synergy	27.8361	52.5288	73.6073	96.8517
With synergy	27.2756	47.0080	63.2642	80.8730
Improvement (%)	2.0136	10.5099	14.0517	16.4981

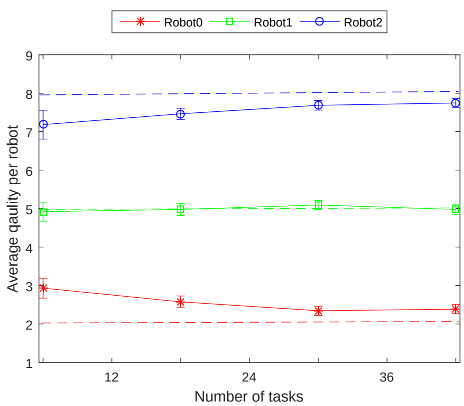
Table 5.2: Percentage of decreasing total traveled distance by applying the proposed online synergy method in WSM auction-based.

The auction-based approach based on weighted sum model gives the minimal quality satisfaction compared to the other two approaches, due to the low weight its associated with this objective. Table 5.3 shows that the average quality requirements of the tasks that are executed by the high quality robot is deviated in negative, which means that the robot executes tasks that are required quality less than the robot quality. In the other hand, the opposite is happened with the robot with the low quality in which its tasks average quality requirements deviated in positive, meaning that the robot performs tasks with higher quality requirements.

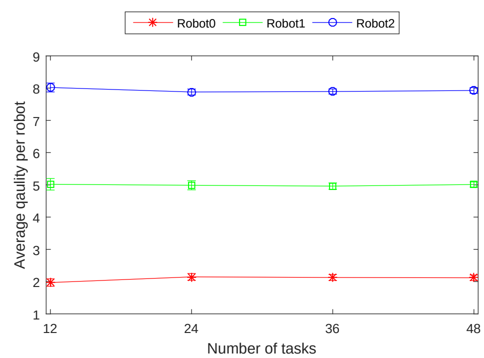
These deviating gets lesser with the increase in number of tasks, because there will be more tasks requires every quality level. Fig.5.5 reveals that the naive auction based produces poor quality as a result of not considering quality in its assignment process.

High	-0.8171	-0.5354	-0.3143	-0.2542
Medium	- 0.0805	-0.0221	+0.0935	-0.0221
Low	+0.9319	+0.5745	+ 0.3852	+0.3436

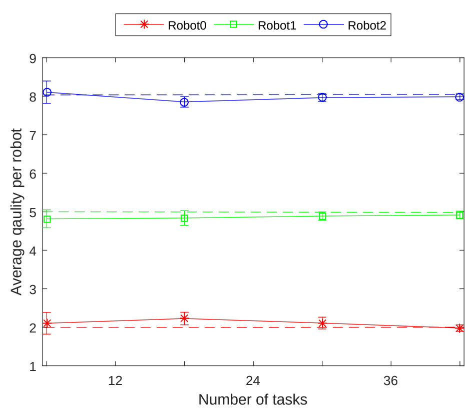
Table 5.3: The deviation from the quality satisfaction using WSM auction-based



(a) Using weighted sum model



(b) Using fuzzy logic rules



(c) Using threshold-based approach

Figure 5.4: Average quality of executed tasks per robot with different number of tasks in WSM auction-based.

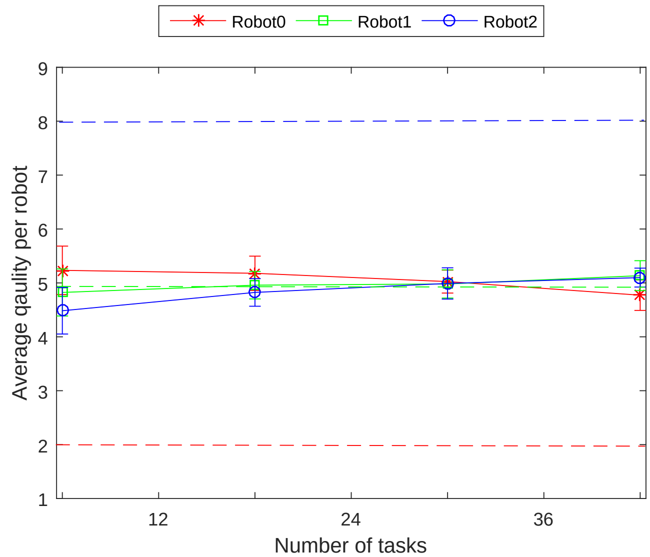
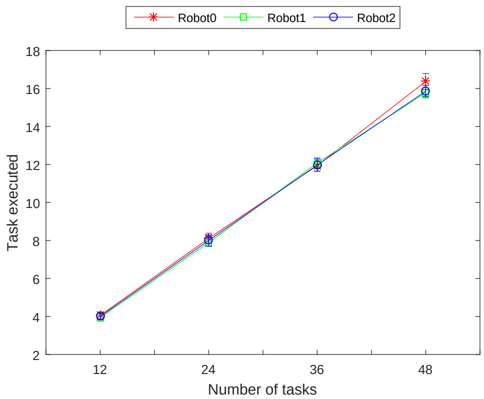


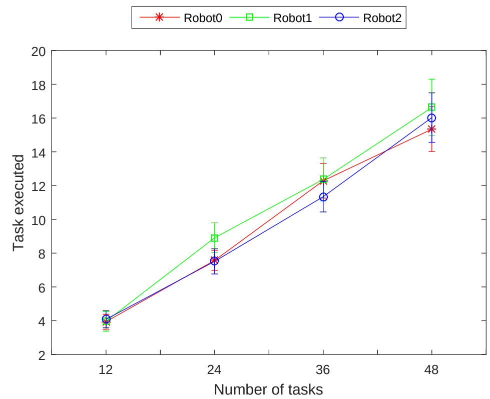
Figure 5.5: Average quality of executed tasks per robot based on distance objective only in WSM auction-based.

5.1.3 Load balancing

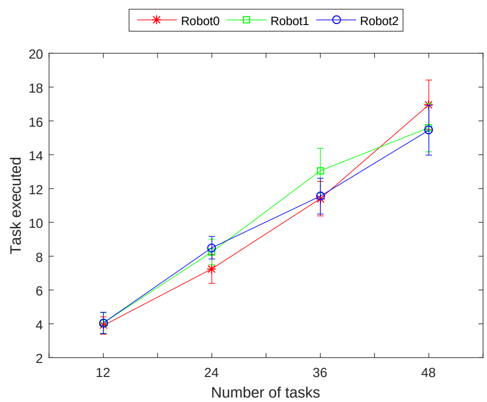
The weighted sum model produces the best load balancing among the three methods almost each robot gets one third of the total number of tasks as shown in Fig.5.6a, which is an expected result because the load balancing objective weight equals to one third of the total weight. Whereas, in fuzzy logic method the load of each robot deviates from the optimal by $\pm 9.4\%$, see Fig. 5.6b. However, threshold based does maintain load balance but not comparable to the one yields by the weighted sum, with deviation on average equals to $\pm 10.62\%$ from the optimal load due to the absent of the explicit coordination in threshold-based approach.



(a) Using weighted sum model



(b) Using fuzzy logic rules

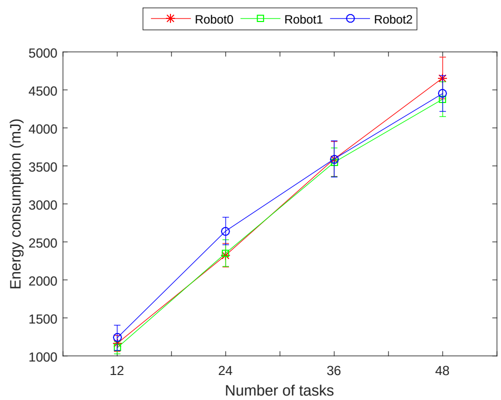


(c) Using threshold based approach

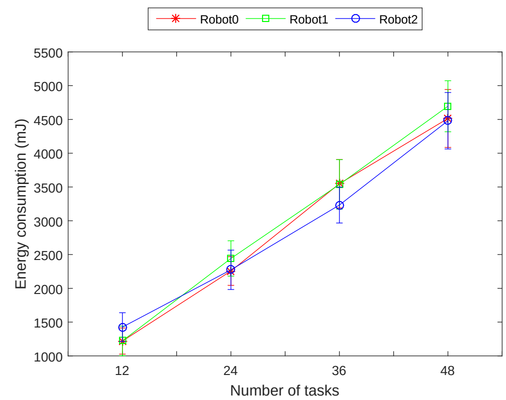
Figure 5.6: Average number of task executed per robot in WSM auction-based.

5.1.4 Energy and resource consumption

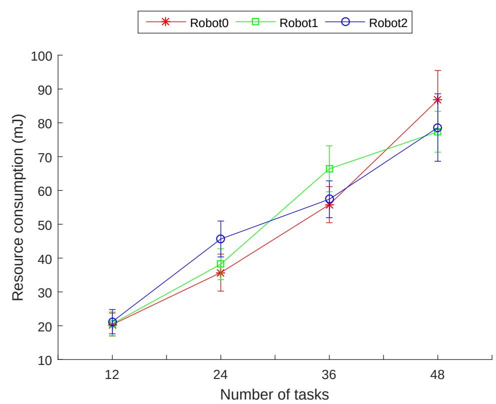
Motion and task execution is the main two sources for energy consumption. Hence, the consumed energy is proportional to the total traveled distance and to the number of accomplished tasks by a robot. A glance at Fig.5.1, 5.6,5.7 shows that energy consumption trend depends on the distance crossed and number of tasks accomplished by a robot. Similarly, resource consumption depends on the number of tasks that are executed by a robot. Comparing the tendency of the number of tasks executed by a robot in Fig.5.6 and the amount of consumed resources in Fig.5.8 reveals the similarity in their trends. For example, the WSM approach produces an optimal load balancing, see Fig.5.6a, and it also produces optimal resource balancing as shown in Fig.5.8a. the same case for threshold-based approach, see Fig.5.6c,5.8c.



(a) Using wiethted sum model

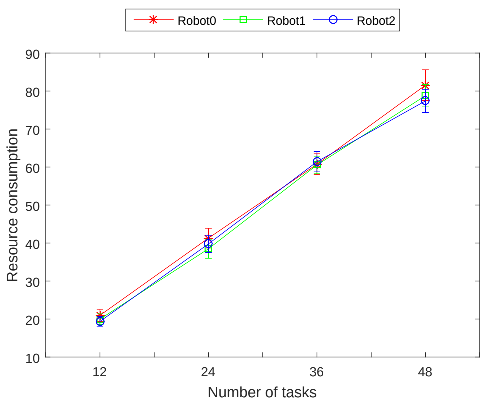


(b) Using fuzzy logic rules

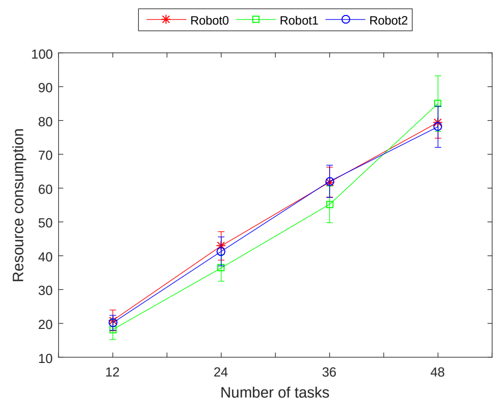


(c) Using threshold based approach

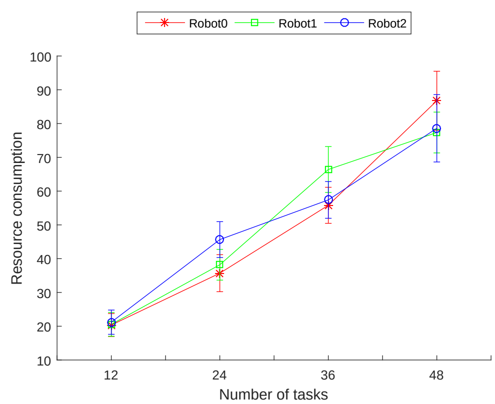
Figure 5.7: Average energy consumption per robot in WSM auction-based.



(a) Using weighted sum model



(b) Using fuzzy logic rules



(c) Using threshold based approach

Figure 5.8: Average resource consumption per robot in WSM auction-based.

5.1.5 Limited communication range

Limiting the range of communication is going to effect the performance of auction-based task allocation approach which utilizes the communication for the auction process. However, for the threshold-based we assumed that distance threshold is always less than communication range, hence it will not get effected by the communication range. Hence, we study the effects of limited communication on the proposed WSM auction-based approach. As Fig.5.9 shows, limited communication (communication range is 1/8 of the diagonal of the area) causes to increase the total traveled distance. This increase is a result of a situation when a suitable robot for a task is not able to bid in the auction because it is out of the auctioneer communication range, yields to assign a task to a robot far from the task. Obviously, the increase in traveled distance is getting larger as the number of tasks getting larger or the communication range getting smaller. Table 5.4 shows the increase in total traveled distance as a percentage with respect to the traveled distance in fully connected network. It shows that almost the increase is by 8%.

Number Tasks	12	24	36	48
Fully connected	39.8163	74.1772	100.2636	128.3988
Limited connection	42.6951	79.3649	108.7632	139.3439
Distance Increase (+%)	7.2301	6.9937	8.4773	8.5243

Table 5.4: Percentage of increase in total traveled distance due to limited connection range in WSM auction-based.

With respect to load balance, although Fig.5.10 shows that load balance in large communication range is better than on short communication range, load balance does not get effected by limiting the communication range that much for

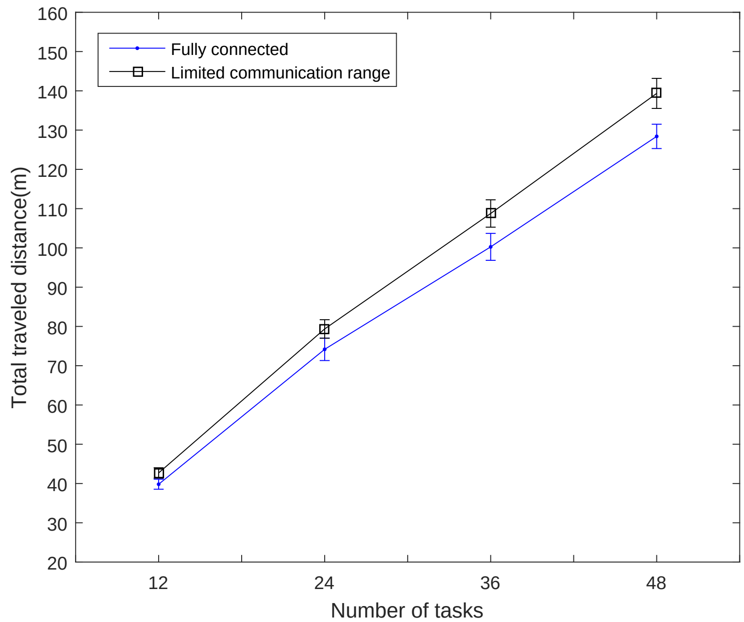
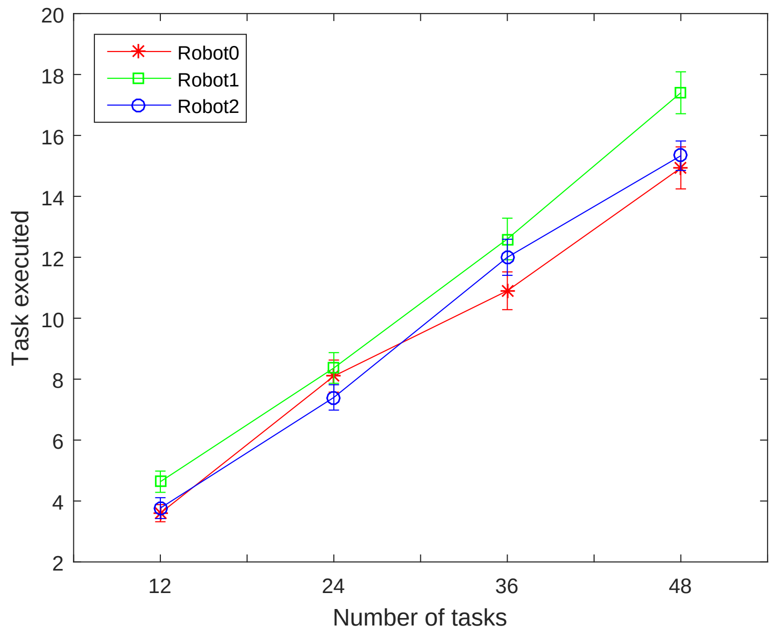


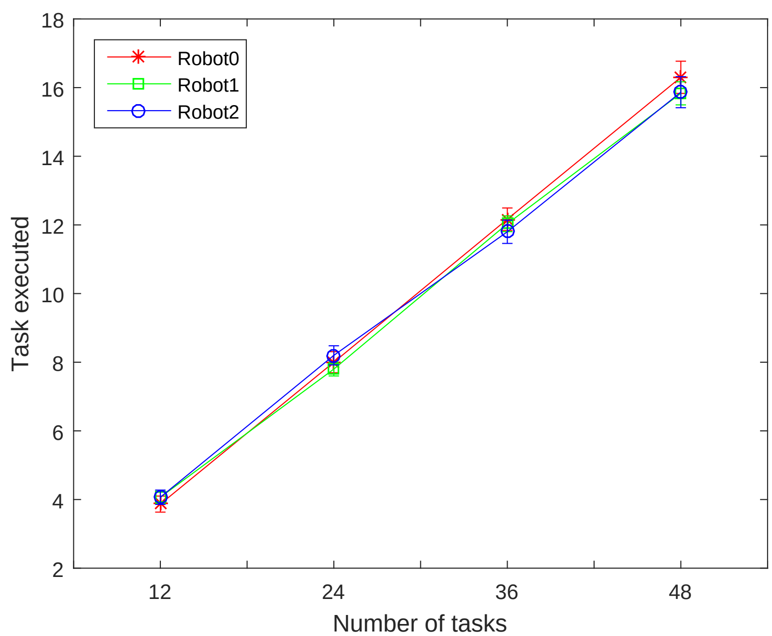
Figure 5.9: Average traveled distance per robot in a limited and full communication range in WSM auction-based.

the reason that tasks appear randomly in the area.

In the case of quality satisfaction, with a limited communication it happens sometimes that there is no robot in the auction proximity that satisfies the task quality, and consequently it leads to assigning the task to a robot which does not satisfy its quality requirement, see Fig.5.11. Table 5.5 illustrates how does the average quality of accomplished tasks for each robot deviate from the optimal quality. For high quality robot ($Robot2$) it the highest set of quality and there are no tasks require quality higher than that. Therefore, when it deviates it performs tasks require lower quality than its quality, and that is why its quality is shifted by negative value. In contrast, low quality robot ($Robot0$) covers the lowest part of tasks quality requirements, therefore when it deviates, it perfroms tasks with higher quality requirements, and that's why its quality average is shifted



(a) Load balance with communication range 1/8 of the diagonal of the area.



(b) Load balance with communication range 1/2 of the diagonal of the area.

Figure 5.10: The effect of limited communication range on load balance for the WSM auction-based.

positively. However, the medium quality robot ($Robot1$) covers the middle part of tasks quality requirements, hence, it deviates sometimes by performing tasks heigher or lower than its quality, eventually the average will keep the same (in the middle), see Table 5.6.

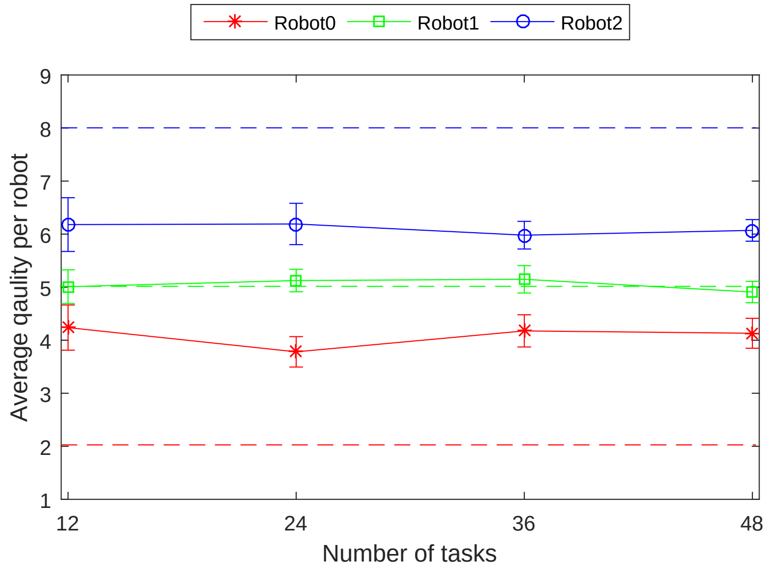
High quality	-1.8204	-1.8088	-2.0783	-1.9296
Medium quality	0.0082	0.1248	0.0449	-0.0919
Low quality	2.2378	1.7809	1.9413	2.1308

Table 5.5: The deviation from the quality satisfaction due to limited communication in WSM auction-based.

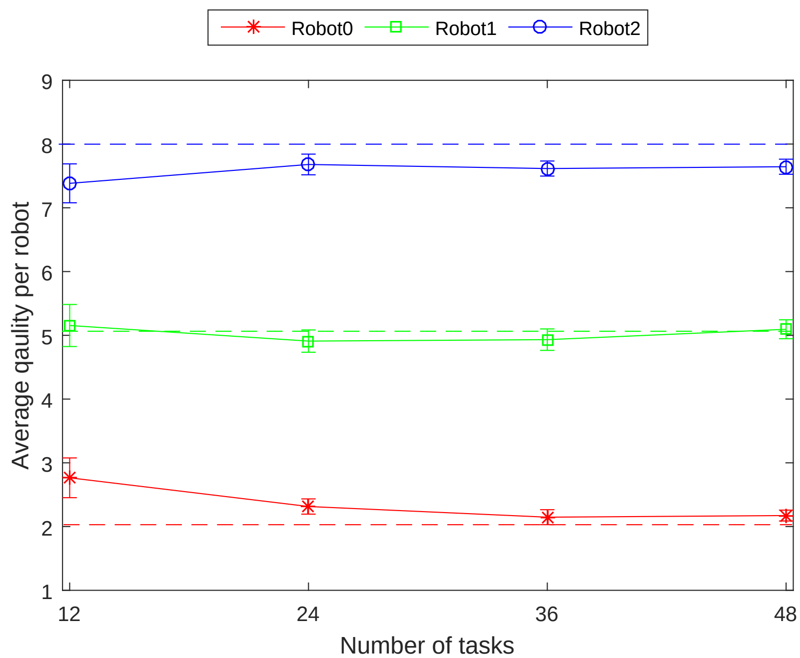
	Task quality																
Robot0	4	5	4	9	3	3	1	3	2	8	8	5	6	1	3	2	1
Robot1	6	4	6	6	6	8	4	4	4	2	6	2	6	8			
Robot2	5	6	4	8	9	9	5	9	8	9	7	6	7	5	1	3	5

Table 5.6: Sample of quality requirements for the tasks assigned to each robot.

Under a limited communication range there is a possibility that a task may not get assigned by the proposed auction based approach. This happens when a robot receives a winning message $winMsg$ from an auctioneer and replies by accept message $accpetMsg$ but then it missed the final Ack message. Please refer to section 3.10 for further details. Fig.5.12 shows how rare this happen, only one task out of 1440 tasks is unallocated if the communication range is quarter the diagonal of the area (18m). It reaches 10 tasks if communication range is too short only 4.5m in an area of 50mx50m.



(a) Average quality of executed tasks with communication range 1/8 of the diagonal of the area in WSM auction-based.



(b) Average quality of executed tasks with communication range 1/2 of the diagonal of the area in WSM auction-based.

Figure 5.11: The effect of limited communication range on quality satisfaction in WSM auction-based.

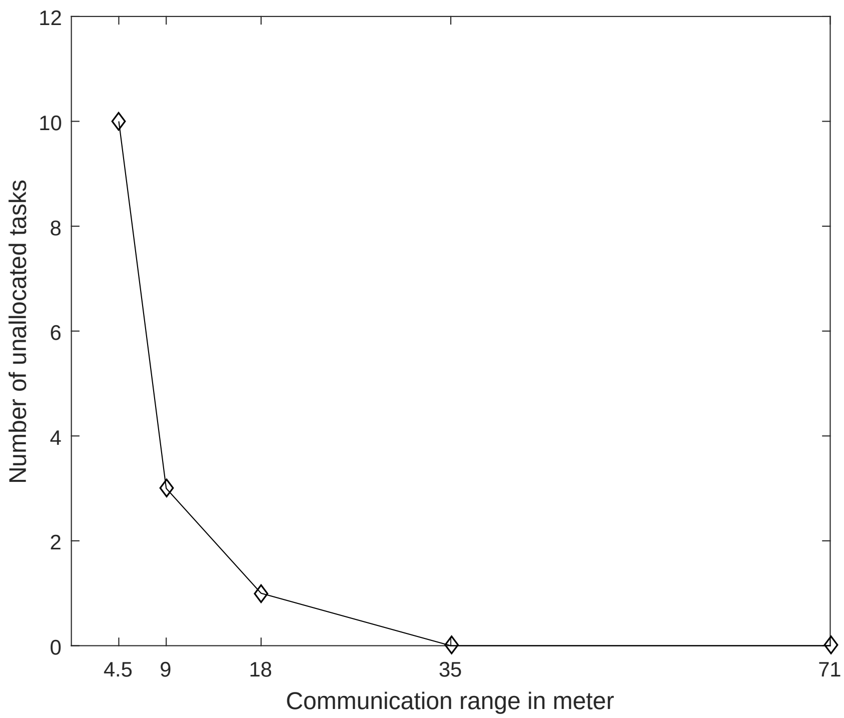


Figure 5.12: Number of unallocated tasks under different communication range (area size is 50mx50m, number of tasks 1440) in WSM auction-based.

5.1.6 Long-term Simulation Run

The long-term experiment has been conducted to show the robustness of the proposed method over a long-run scenario. We have tested the proposed method for one long run with 300 tasks into a different area size. The rest of experiment parameters are kept the same as in Table 4.1. We assume each robot has enough resources to performs all assigned tasks.

In Fig.5.13 the results of the load balance among robots with total tasks of 300 the robots get about the same number of tasks. Even when area size changed still the proposed method is able to assign the load equally with a relevant small deviation due to the contradict between load balance and the other two objectives i.e. minimizing traveled distance and maximizing quality satisfaction.

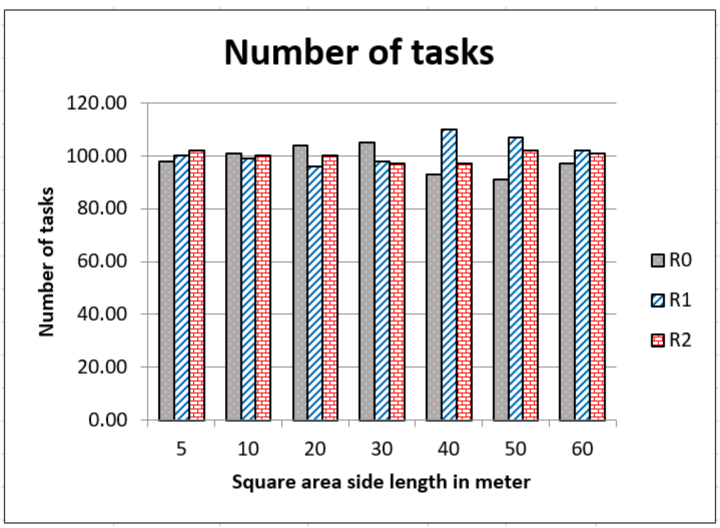


Figure 5.13: Number of tasks executed per robot in long run (300 tasks) with difference area size in WSM auction-based.

Total traveled distance per robot is shown in Fig.5.14. Traveled distance increases as the area size increases, and roughly the robots have traveled the same distance in each experiment within the same area size.

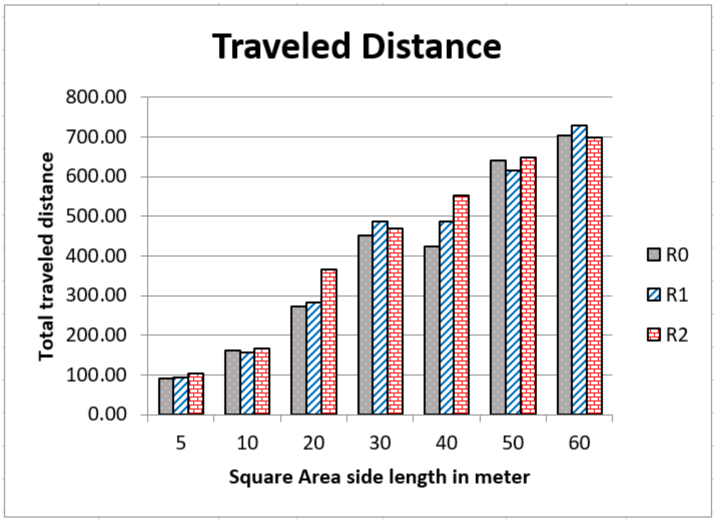


Figure 5.14: Traveled distance for long run (300 tasks) with difference area size in WSM auction-based.

Similar to workload balance, quality satisfaction does not get affected by experiment area size. As Fig.5.15 illustrates, the average quality for tasks assigned to a robot is almost the same quality of the robot itself, which means that the robot performs tasks with quality equal or close to its quality level requirements.

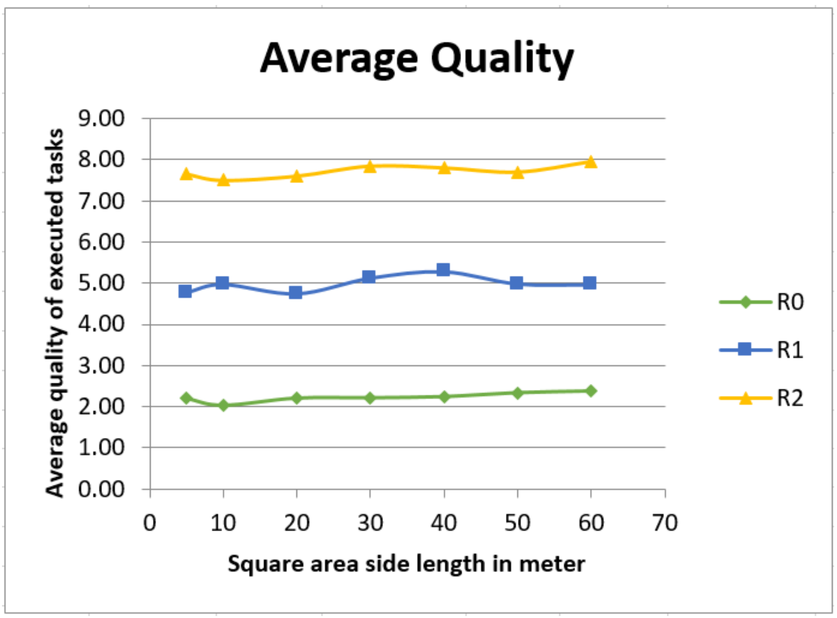
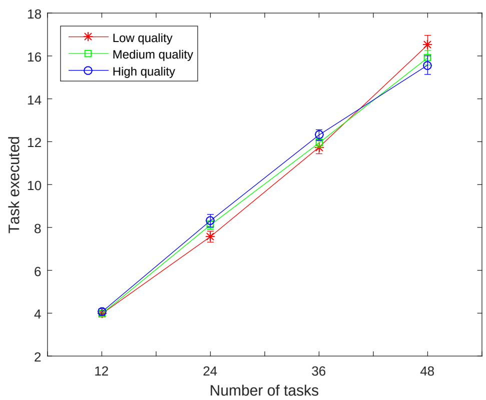


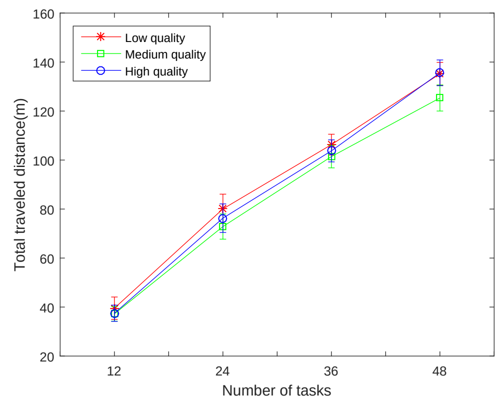
Figure 5.15: Average quality tasks executed per robot in long run (300 tasks) with difference area size in WSM auction-based.

5.1.7 Scalability test

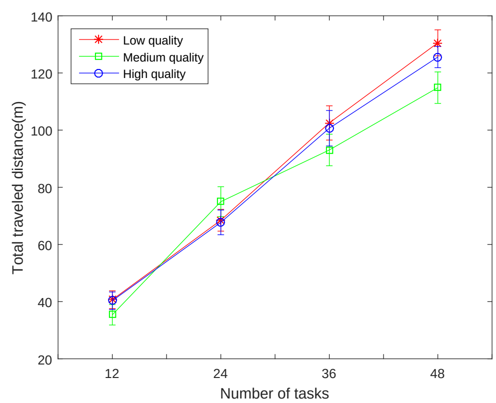
We have tested the scalability of the proposed multi-objective auction based approach by using more than three robots (i.e. 6, 9, and 12 robots). We found that the proposed auction based approach performs well and works in a stable manner by achieving the objectives; minimizing total traveled distance, load balance, and satisfy tasks quality requirements.



(a) In case of 6 robots.



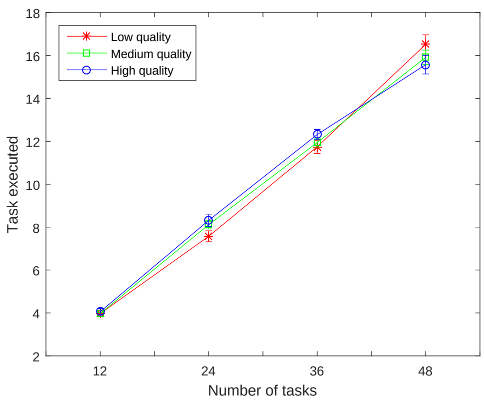
(b) In case of 9 robots.



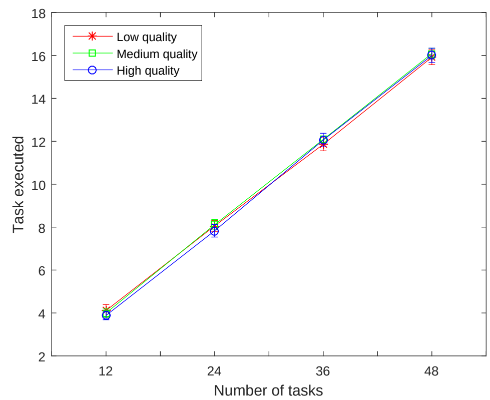
(c) In case of 12 robots.

Figure 5.16: Total traveled distance per robot with varying number of tasks (area size =50mx50m) in WSM auction-based.

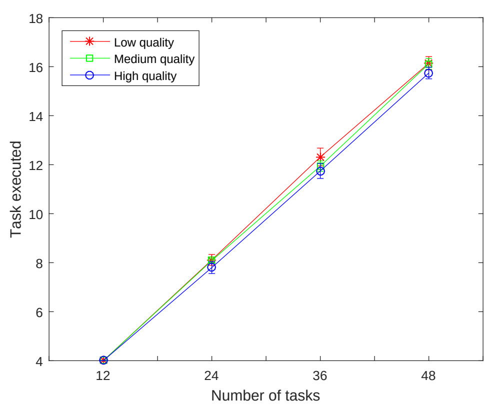
The proposed auction based approach maintains almost same traveled distance for all robots even when number of robots scaled by double as in Fig.5.16a, triple in Fig.5.16b, and quadruple in Fig.5.16c. Moreover, the average total traveled distance per robot decreased by half whenever the number of robots doubled because the load of one robot will be divided between two robots when we double the number of robots. For instance, on average the traveled distance in case of 6 robots in Fig.5.16a at 48 tasks is almost 70m, and in Fig.5.16c where the number of robots is 12, the average traveled distance in 48 tasks is almost 33m.



(a) With 6 robots



(b) With 9 robots



(c) With 12 robots

Figure 5.17: Average number of task executed per robot with different number of robots and tasks in WSM auction-based.

As shown in Fig.5.17, proposed method successfully distributed the load equally among all available robots such that a robot load is equal to the number of tasks divided by the number of available robots.

We have tested quality satisfaction for more than three robots as follows: we keep the number of robots quality at three levels (low, medium, and high are represented by numbers 2, 5 and 8 respectively), and we increased number of robots that have same quality level; starting with 2 robots for each quality level, up to 4 robots for each quality level. Table 5.7 shows the robots' IDs with their associated quality level.

	Robot quality level		
	Low	Medium	High
Robot ID	0	1	2
	3	4	5
	6	7	8
	9	10	11

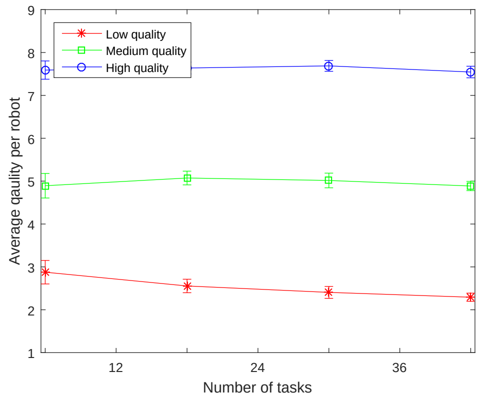
Table 5.7: Robots ID and associated quality level.

Fig.5.18 reveals that tasks have been assigned to propitiate robots, or in another word, low quality robots perform low quality tasks and medium quality robot perform medium quality robot and so on. Hence, as Fig.5.18 the average quality of executed task for each robot is almost equal to its own quality level.

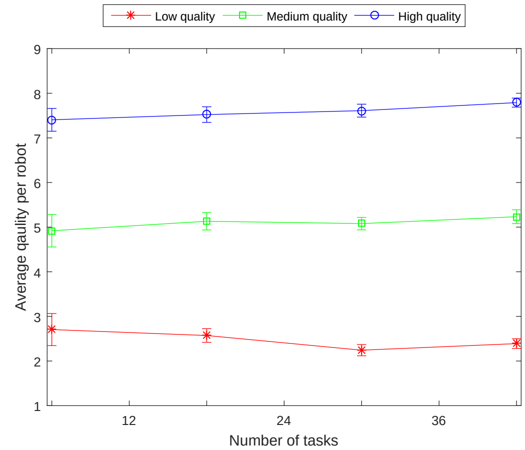
5.2 Total number of messages analysis

Lemma 5.1 *The total number of messages in the proposed auction based approach is given by O(mn), where m is the number of robots and n is the number of tasks.*

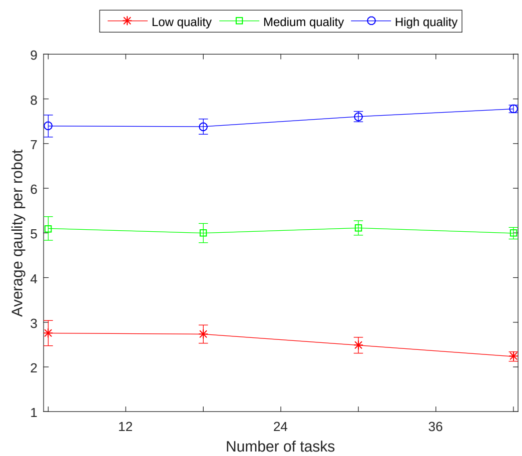
Proof: For each task an auctioneer broadcasts one message and it gets reply from all other robots except itself (m-1) (This will happen in the worst case if the



(a) Using 6 robots, 2 at each of the three quality level.



(b) Using 9 robots, 3 at each of the three quality level.



(c) Using 12 robots, 4 at each of the three quality level.

Figure 5.18: Average quality of executed tasks per robot with different number of tasks in WSM auction-based.

network is fully connected). Finally the auctioneer replies by one message to the winner robot. So the total number of messages for each task are equal to one auction message plus (m-1) biding messages plus one winning message (i.e. 1+(m-1)+N).

Lemma 5.2 *The total number of messages in the proposed threshold based ap-*

proach is given by O(nm), where m is the number of robots and n is the number of tasks.

Proof: Basically threshold based does not utilize message communication except for solving the tie problem, hence the worst case if tie problem has appeared for all n number of tasks, in this case for each task there will be one message and then the total messages are m*n messages.

5.3 Experimental setup and results

We have validated the proposed method through a real test bed experiment using three Turtlebot2 robots (see Fig.[2]). In this section, we present the robot platform, experiment environment setup and the results along with detailed discussions.

5.3.1 Robot platform

We used a team of three Turtelbot2 mobile robots in our experimental test bed. The Turtelbot2 is a 354 x 354 x 420 mm two-wheeled mobile base platform. It comes with a 3D sensor for obstacle avoidance and notebook which used as an interface for communicating robot's mobile base. MOTA is implemented on the Turtlebot2 notebook, and it communicates with others robots via wireless (802.11).

5.3.2 Experiment setup

In order to evaluate the proposed method, a real experiment has been conducted using same the scenarios we have used in the simulation. We have used three Turtlebot2 robots denoted as $(R^0, R^1, and\ R^2)$; each has sufficient energy and virtual resources to accomplish assigned tasks. As we did in the Webots simulator robots have three quality levels; low, medium and high, respectively. Initially, we have specified the locations for 15 tasks in the experiment area ($5m\ x\ 5m$), where a task can appear randomly in any of these locations. Fig.5.19 shows robots location and tasks location at the beginning of the experiment.

The Turtlebot2 robot has a 3D camera which is used to draw a map for the area to be used in navigation later. We have mapped the task real-world location into robot's map coordination; hence, a robot bids based on its location and task location in its map.

A laptop used to generate tasks randomly in the area over the time and sends an advertisement message to a randomly selected robot which will be the auctioneer for that task. At the end of each run, robots send a statistical data to the laptop including their total traveled distance, a number of assigned tasks, resource consumptions and the average quality satisfaction. Video footage of these experiments is available at https://www.youtube.com/watch?v=SC-V6tRVdIo.

The experiment has been repeated 11 times, in each of which 21 tasks have been generated with an exponential inter-arrival time of mean equal to 100*sec*. Average robot task execution time has been set to be 230*sec*. The results collected and displayed with a confidence level of 90%.

5.3.3 Experimental results

We have observed that the proposed method successfully distributes tasks among available robots almost equally, as it is shown in Fig.5.21. Consequently, robots travel the same distance in average as it is shown in Fig.5.22 where the confidence interval of the average total traveled distance for robots are overlapped. Resources have been consumed equally from all robots; which means no robot run out of

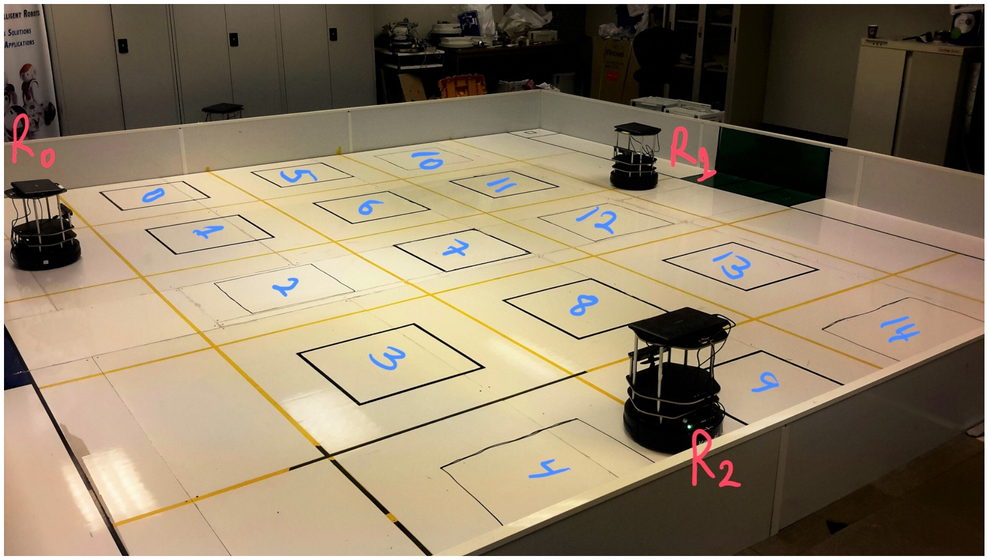


Figure 5.19: Experiment environment with three Turtelbot2 ($R_0, R_1, and R_2$), and 15 locations where a task possibly can appear.

resources before other. In other words, the robotic network continues providing all available level of qualities provided by the three type of robots.

Fig.5.23 shows that the average quality level of assigned tasks for each robot is closed to a robot quality level. Which means all tasks assigned to a robot that satisfies their required quality level.

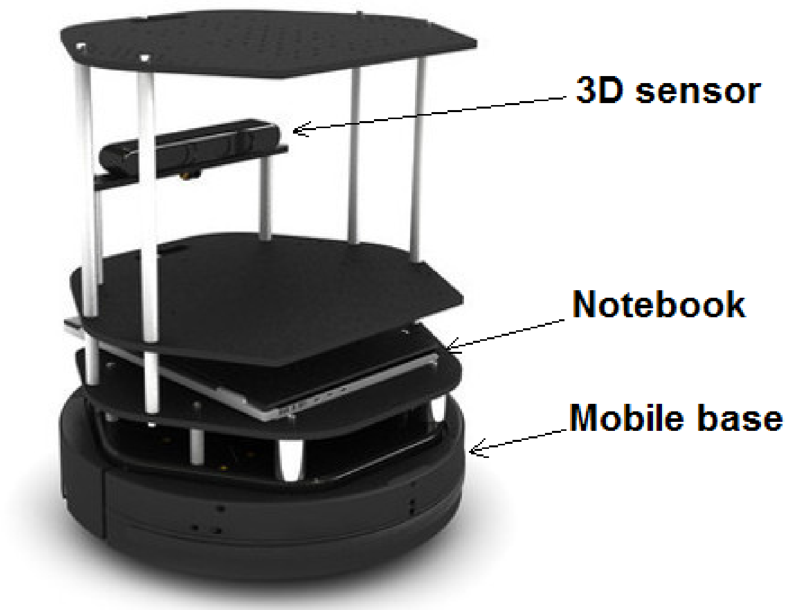


Figure 5.20: A close look at Turtlebot2 [2].

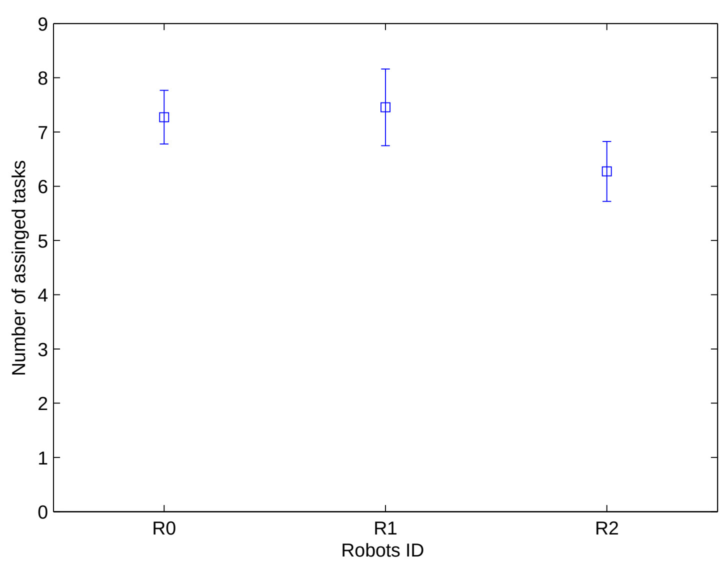


Figure 5.21: Average number of tasks assigned to each Turtlebot2 robot

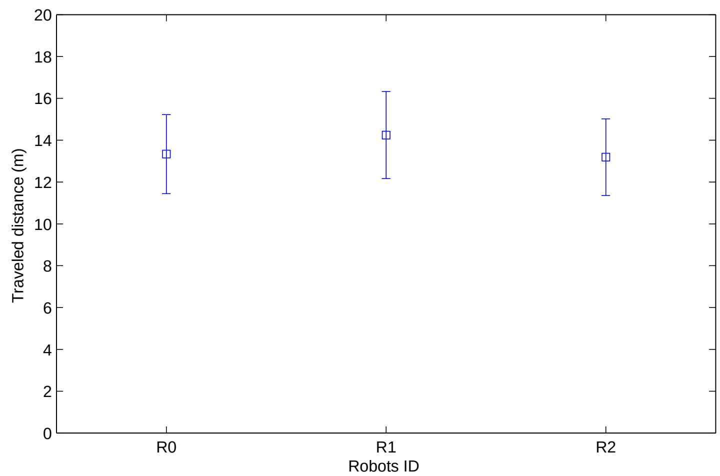


Figure 5.22: Average total travelled distance per each Turtlebot2 robot

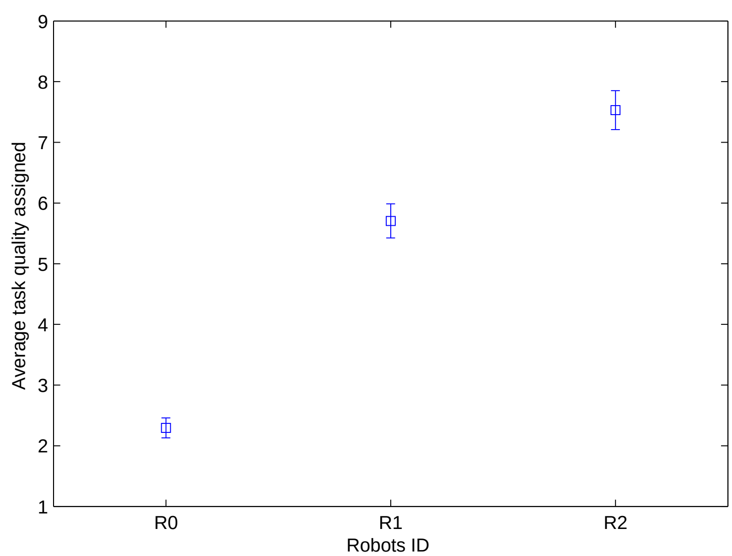


Figure 5.23: Average assigned task quality per Turtobot2 robot. Quality level low(2), medium(5), and high (8) are robots R^0, R^1, R^2 quality level respectively.

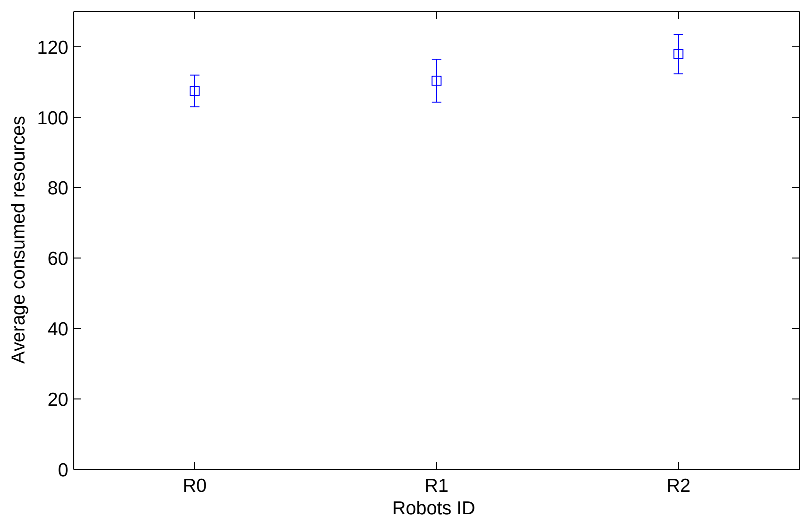


Figure 5.24: Average resource consumptions per robot

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 Conclusion

We proposed a distributed multi-objectives task allocation approach as a solution for multi-robot task allocation. The proposed approach targets MRTA problem in a context where tasks dynamically appear during the mission and there is no a priori information about the location and the requirements of the tasks, as well as the position and available resources in the robots, are changing over the time.

The proposed method is an auction-based approach where a team of robots cooperates explicitly in order to assign a task in a distributed fashion. They use the auction concept for cooperation , such that in each auction, robots bid for an announced task based on their local information (location, quality, and load). Two methods have been used for combining these factors into one scalar value that represents the fitness of the robot. The first method uses the weighted sum model in which the factors has been summed up with appropriate weights. The second method utilizes the fuzzy logic system, where each factor is considered as an input variable for the fuzzy system, and the output variable is the corresponding fitness value. Moreover, we have investigated the performance of threshold-based paradigm.

The proposed approaches have been tested extensively on a simulated robotics network using Khepera-III robots on Webots simulator. We also demonstrated the proposed methods using Turtlebot2 robots. We found that the results from the simulation and real experiments have the same trend, which implies the usefulness and practicality of the proposed method in real world scenarios.

We have shown that the travel distance, load, and quality satisfaction objectives are independent while the rest objectives such robot resources and remaining energy can be satisfied if the former objectives are satisfied. This interesting result minimizes the dimension of our problem. Moreover, our proposed approaches were able to achieve the desired assignment, and consequently minimize the total travel distance per robot, distribute the load equally among robots, and satisfy the task quality requirement.

6.2 Future work

As a future work, the proposed method can be extended to form a framework, which includes, besides the factors we already consider, task waiting time and task priority, as well as, testing the proposed method under different scenarios where tasks appear not uniformly in the area e.g. when they emerge in a punch pattern. Also studying the proposed method with different initial energy and resources is one future direction for this work. In our study, we assume one task with different quality levels, this can be extended into multi-task with multi-quality levels scenarios. The concept of synergy can be extended to not only consider the spatial synergy between tasks but also the type and quality level of the tasks.

REFERENCES

[1] A. Koubâa and A. Khelil, *Cooperative Robots and Sensor Networks 2015*. Springer, 2015, vol. 604.

[2] I. Open Source Robotics Foundation, "Turtlebot2," 2016. [Online]. Available: http://www.turtlebot.com/

[3] I. I. F. of Robotics, "Industrial robot statistics," 2014. [Online]. Available: http://www.ifr.org/industrial-robots/statistics/

[4] M. P. Singh, "Multiagent systems," *Multiagent Systems: A Theoretical Framework for Intentions, Know-How, and Communications*, pp. 1–14, 1994.

[5] U. Baroudi, G. Sallam, M. Al-Shaboti, and M. Younis, "Gps-free robots deployment technique for rescue operation based on landmark's criticality," in *Wireless Communications and Mobile Computing Conference (IWCMC), 2015 International*. IEEE, 2015, pp. 367–372.

[6] S. Bhattacharya, R. Ghrist, and V. Kumar, "Multi-robot coverage and exploration on riemannian manifolds with boundaries," *The International Journal of Robotics Research*, vol. 33, no. 1, pp. 113–137, 2014.

[7] M. B. Dias and A. Stentz, "Traderbots: A market-based approach for resource, role, and task allocation in multirobot coordination," *CMU*, 2003.

[8] B. U. Sallam G., "Cover: A cooperative virtual force robot deployment technique," in *Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing (CIT/IUCC/DASC/PICOM), 2015 IEEE International Conference on.* IEEE, 2015, pp. 1308–1315.

[9] J. Vertut, *Teleoperation and robotics: applications and technology.* Springer Science & Business Media, 2013, vol. 3.

[10] R. Pfeifer, H. G. Marques, and F. Iida, "Soft robotics: the next generation of intelligent machines," in *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence.* AAAI Press, 2013, pp. 5–11.

[11] H. Ardiny, S. Witwicki, and F. Mondada, "Construction automation with autonomous mobile robots: A review," in *Robotics and Mechatronics (ICROM), 2015 3rd RSI International Conference on.* IEEE, 2015, pp. 418–424.

[12] F. Basoeki, F. Dalla Libera, E. Menegatti, and M. Moro, "Robots in education: New trends and challenges from the japanese market," *Themes in Science and Technology Education*, vol. 6, no. 1, pp. 51–62, 2013.

[13] F. A. Auat Cheein and R. Carelli, "Agricultural robotics: Unmanned robotic service units in agricultural tasks," *Industrial Electronics Magazine, IEEE*, vol. 7, no. 3, pp. 48–58, 2013.

[14] W. Burgard, M. Moors, C. Stachniss, and F. E. Schneider, "Coordinated multi-robot exploration," *Robotics, IEEE Transactions on*, vol. 21, no. 3, pp. 376–386, 2005.

[15] M. B. Dias, R. Zlot, N. Kalra, and A. Stentz, "Market-based multirobot coordination: A survey and analysis," *Proceedings of the IEEE*, vol. 94, no. 7, pp. 1257–1270, 2006.

[16] M. J. Krieger and J.-B. Billeter, "The call of duty: Self-organised task allocation in a population of up to twelve mobile robots," *Robotics and Autonomous Systems*, vol. 30, no. 1, pp. 65–84, 2000.

[17] W. Agassounon and A. Martinoli, "Efficiency and robustness of threshold-based distributed allocation algorithms in multi-agent systems," in *Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 3*. ACM, 2002, pp. 1090–1097.

[18] B. P. Gerkey and M. J. Matarić, "A formal analysis and taxonomy of task allocation in multi-robot systems," *The International Journal of Robotics Research*, vol. 23, no. 9, pp. 939–954, 2004.

[19] R. Smith, "Communication and control in problem solver," *IEEE Transactions on Computers*, 1980.

[20] S. Koenig, C. Tovey, M. Lagoudakis, V. Markakis, D. Kempe, P. Keskinocak, A. Kleywegt, A. Meyerson, and S. Jain, "The power of sequential single-item auctions for agent coordination," in *Proceedings of the National Conference*

on *Artificial Intelligence*, vol. 21, no. 2. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2006, p. 1625.

[21] K. L. Hoffman, M. Padberg, and G. Rinaldi, "Traveling salesman problem," in *Encyclopedia of Operations Research and Management Science*. Springer, 2013, pp. 1573–1578.

[22] Z. Mi, Y. Yang, H. Ma, and D. Wang, "Connectivity preserving task allocation in mobile robotic sensor network," in *Communications (ICC), 2014 IEEE International Conference on*. IEEE, 2014, pp. 136–141.

[23] N. Kalra and A. Martinoli, "Comparative study of market-based and threshold-based task allocation," in *Distributed autonomous robotic systems 7*. Springer, 2006, pp. 91–101.

[24] B. P. Gerkey and M. J. Matari, "Sold!: Auction methods for multirobot coordination," *Robotics and Automation, IEEE Transactions on*, vol. 18, no. 5, pp. 758–768, 2002.

[25] D.-H. Lee, S. A. Zaheer, and J.-H. Kim, "A resource-oriented, decentralized auction algorithm for multirobot task allocation," *Automation Science and Engineering, IEEE Transactions on*, vol. 12, no. 4, pp. 1469–1481, 2015.

[26] W. Sheng, Q. Yang, J. Tan, and N. Xi, "Distributed multi-robot coordination in area exploration," *Robotics and Autonomous Systems*, vol. 54, no. 12, pp. 945–955, 2006.

[27] M. Elango, S. Nachiappan, and M. K. Tiwari, “Balancing task allocation in multi-robot systems using k-means clustering and auction based mechanisms,” *Expert Systems with Applications*, vol. 38, no. 6, pp. 6486–6491, 2011.

[28] J. Gong, J. Qi, G. Xiong, H. Chen, and W. Huang, “A ga based combinatorial auction algorithm for multi-robot cooperative hunting,” in *Computational Intelligence and Security, 2007 International Conference on*. IEEE, 2007, pp. 137–141.

[29] A. T. Tolmidis and L. Petrou, “Multi-objective optimization for dynamic task allocation in a multi-robot system,” *Engineering Applications of Artificial Intelligence*, vol. 26, no. 5, pp. 1458–1468, 2013.

[30] S. K. Seelan, S. Laguette, G. M. Casady, and G. A. Seielstad, “Remote sensing applications for precision agriculture: A learning community approach,” *Remote Sensing of Environment*, vol. 88, no. 1, pp. 157–169, 2003.

[31] A.-K. Mahlein, E.-C. Oerke, U. Steiner, and H.-W. Dehne, “Recent advances in sensing plant diseases for precision crop protection,” *European Journal of Plant Pathology*, vol. 133, no. 1, pp. 197–209, 2012.

[32] X. Zheng, S. Koenig, and C. Tovey, “Improving sequential single-item auctions,” in *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*. IEEE, 2006, pp. 2238–2244.

[33] D. S. Johnson and L. A. McGeoch, "The traveling salesman problem: A case study in local optimization," *Local search in combinatorial optimization*, vol. 1, pp. 215–310, 1997.

[34] W. H. Lenagh, *Multi-robot task allocation: a spatial queuing approach.* UNIVERSITY OF NEBRASKA AT OMAHA, 2013.

[35] L. Liu and D. A. Shell, "Large-scale multi-robot task allocation via dynamic partitioning and distribution," *Autonomous Robots*, vol. 33, no. 3, pp. 291–307, 2012.

[36] E. Triantaphyllou, *Multi-criteria decision making methods: a comparative study.* Springer Science and Business Media, 2013, vol. 44.

[37] R. E. Bellman and L. A. Zadeh, "Decision-making in a fuzzy environment," *Management science*, vol. 17, no. 4, pp. B–141, 1970.

[38] L. A. Zadeh, "Fuzzy sets," *Information and control*, vol. 8, no. 3, pp. 338–353, 1965.

[39] Webots, "http://www.cyberbotics.com," commercial Mobile Robot Simulation Software. [Online]. Available: http://www.cyberbotics.com

[40] KheperaIII, "http://www.k-team.com," k-team Mobile Robotics. [Online]. Available: http://www.k-team.com

[41] A. O'Dwyer, *Handbook of PI and PID controller tuning rules.* World Scientific, 2009, vol. 57.

Vitae

- Name: Mohammed Mahmood Ali Al-Shaboti
- Nationality: Yemeni
- Date of Birth: 05/07/1986
- Email: *alshaboti.it@gmail.com*
- Permenant Address: Taiz, Yemen